-64-194

(FINAL REPORT)

QUANTITATIVE METHODS FOR INFORMATION PROCESSING SYSTEMS EVALUATION

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-194

JANUARY 1964

Alan Taylor
John R. Hillegass
Norman Statland

DIRECTORATE OF COMPUTERS
DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 1134

(Prepared under Contract No. AF 19 (628)-2838 by the AUERBACH Corporation, Philadelphia 3, Penna.)

ESD-TDR-64-194

(FINAL REPORT)

QUANTITATIVE METHODS FOR INFORMATION PROCESSING SYSTEMS EVALUATION

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-194

JANUARY 1964

Alan Taylor
John R. Hillegass
Norman Statland

DIRECTORATE OF COMPUTERS
DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 1134

(Prepared under Contract No. AF 19 (628)-2838 by the AUERBACH Corporation, Philadelphia 3, Penna.)

FOREWORD

This report was prepared by the Information-Management Sciences Division of the AUERBACH Corporation under United States Air Force Contract Number AF 19(628)-2838. The contract was initially based upon the work related to estimating computer system performance exhibited in AUERBACH Standard EDP Reports. The work was performed by the staff of AUERBACH Standard EDP Reports, under the direction of its Executive Editor, Mr. Norman Statland, from March 1963 through January 1964.

QUANTITATIVE METHODS FOR INFORMATION PROCESSING SYSTEMS EVALUATION

ABSTRACT

This research was directed toward determining whether computer system timings could be simply estimated, with reasonable accuracy, using only standardized descriptions of a computer system and an application. In the course of the research, determination of those elements of a computer and associated programs that are critical to overall timing were to be indicated. This research demonstrates that the approach is feasible. The conclusion is supported by the development of the VECTOR process for readily estimating computer processing times. The system permits prediction of time estimates to process a given application on a specific computer within ± 15%. Preparation of the standard description for a typical computer requires 25 man-hours and for a typical application 2 man-hours, but this needs to be done only once to permit many analyses. The approach does take limitations into account (e. g. , input-limited, computer-limited runs, etc.). Details of the VECTOR process, of the development of specific vectors (standard descriptions) for Honeywell 800, IBM 7074, and UNIVAC III, and of the tests are reported in the supporting technical notes.

REVIEW AND APPROVAL

This technical documentary report has been reviewed and is approved.

Frank E. Herin

FRANK E. HERIN
1st Lt, USAF
Project Engineer
Computer Division

Joseph J. Rosa

JOSEPH J. ROSA
Lt Col, USAF
Acting Director
Directorate of Computers
Deputy for Engineering & Technolgy

KEY WORD LIST

1. COMPUTERS
2. DESIGN
3. TEST SETS
4. MODELS (SIMULATION)
5. VECTOR PROCEDURE
6. PROGRAMMING
7. LANGUAGE
8. MACHINE TRANSLATION
9. DOCUMENTATION
10. DATAPROCESSING SYSTEMS
11. DATA TRANSMISSION SYSTEMS
12. DATA STORAGE SYSTEMS
13. INFORMATION RETRIEVAL
14. COMPUTER SYSTEMS EVALUATION
15. FEASIBILITY STUDIES

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1. INTRODUCTION

1.1 OBJECTIVES

Computers are flexible, complex devices which are being used for a wide variety of applications. Further, there are a large number of computer systems, each with a variety of configurations, available to the user. It is, therefore, important to investigate and develop methods of evaluating different computers and computer configurations for use in specific tasks. The problem of evaluating computers arises frequently in any large organization such as the United States Air Force. Any insights which simplify the evaluating process can have a major effect upon reducing the cost of implementing the computer system.

In recognition of these considerations, the Electronic Systems Division of the Air Force Systems Command undertook to supply research in "Quantitative Methods for Information Processing Systems Evaluation" (ESD PR ES-3-GEN-403-5, 29 October 1962).

The general purpose of this research was to develop insights into the characteristics of specific computers that make them effective for a certain class of job; in particular, magnetic tape data processing applications. Such research should ultimately lead to:

(1) A method of reducing the cost of making specific computer evaluations, (while providing a wider scope in a reduced time period),

(2) Improving computer system design by offering a means to better understanding of the computer characteristics required by the problem,

(3) Standardization of languages and methods involved in specifying computer applications and computer equipment,

(4) A means for providing a prospective user of data processing equipment with an indication of those areas in the program processing that offer potential difficulties for implementation.

The bid specification requested research into the following three specific areas: the definition of functional characteristics of data processing systems; procedures for preparing function vectors (e.g., for evaluating functional characteristics) for tape computations; and procedures for preparing engineering vectors for a class of commercially available computers.

AUERBACH Corporation had already been active in similar research areas as a result of its consulting work in the evaluation of computers for specific applications, and particularly in the preparation of material for the AUERBACH Standard EDP Reports.

The normal procedure in evaluating a computer for a particular application is for the analyst to visualize the way he would actually program the application for the particular computer, taking advantage of the various characteristics of the computer with which he may be acquainted. He estimates the times which will be required by the computer to perform various parts of the processing and then arrives at an overall estimate of performance based on both processing time and the required internal storage capacity. The procedure assumes that a single analyst can be knowledgable (in depth) of both the computer system and the application requirements. In our project, we state that the analyst who specified the problem cannot be the same person who should specify the performance variables of each and every computer system under consideration. Specifically, each specification of the engineering characteristics and derivation of the Engineering Vector should be developed by an experienced programmer from the equipment manufacturer's staff, since only such an individual is capable of preparing a valid engineering vector.

The underlying assumptions of this particular research are that a set of characteristics, when standardized and processed into a series of elements called the Functional Vector[1], can be developed to describe a given class of problems in a concise way, and that another set of characteristics, when standardized and processed into a series of elements called the Engineering Vector[2], can be developed to characterize a computer system configuration and its performance variables in a standard way. Then the proper processing of a particular Functional Vector and a particular Engineering Vector will produce a measure of the performance for the corresponding computer configuration on the particular problem, as described in the Functional Specifications.

---

1 See Appendix II.

2 See Appendix I.

The procedures for evaluation are complicated by the fact that no one version of the program process for a particular application is equally suitable for all computer systems; each of which have different processing capabilities and special features. Moreover, if a particular version is adopted, it would tend to have a bias toward certain computers. For example, a statement of the blocking factor for a program creates a bias. Furthermore, the amount of the bias would be variable and unknown, thus preventing a simple correction of this bias. It was discovered that it was possible to develop a small set of program procedure approaches to the same application problem which could be used to evaluate the performance for most standard computers. It is this approach which the research follows in detail.

In general, the following underlying assumptions are made throughout this research. The prospective user of the system has the responsibility for defining the computer runs, for performing detailed system analysis, and for providing the basic Functional Specifications. The evaluation procedures concentrated on measuring the performance of a computer for a single run; for example, the most critical run within the application. The goal was to choose characteristics (or elements of the vectors) which led to useful evaluation procedures, since there seems to be no other way to arrive at suitable characteristics.

1.2 SUMMARY

1.2.1 The VECTOR Process

The VECTOR process is a new procedure for producing quantitative, objective estimates of the performance of digital computer systems. The essence of the VECTOR process is the production of descriptions, in standardized formats, of the characteristics of each problem to be solved and each computer system to be considered. Furthermore, the method provides a series of synthetic measures of application unknowns, which when adapted to the variations in each computer system design enable the subjective considerations of system evaluations to be removed. Then, by means of straightforward calculations, the estimated processing time for a specific application on a specific computer configuration can readily be produced through a standard procedure. Thus, valid comparisons of applications for varied computer systems can be made, without fear of one system being favored over the other.

These two attributes, together with the actual task, therefore, provide justification for the title VECTOR, which has been applied to the method. In detail

V for VALID,	(in that it estimates computer performance after the application has been adjusted to suit the computer-thus avoiding all kinds of bias.)
E for "ESTIMATING of"	
C for COMPUTER	(it could read COMPARABLE, as two different analysts using the process independently should arrive at strictly comparable results.)
T for TIMING	
O for OBTAINABLE	
R for READILY	(as someone in the field, with a couple of hours' work and a desk calculator, can use the method oncc set up.)

"VECTOR" is also most appropriate as the method uses independent sets of numbers (Vectors) to represent the important characteristics of each computer and each application. Thc Vectors are used both to establish and time thc optimum method on a specific computer system. The Vectors themselves are normally self-sufficient and no other knowledge of either the problem or the computer is assumed.

In these circumstances it can be seen that the creation of these Vectors is an unusually responsible process. This document is concerned with the preparation of the Computer Vector, shows the procedural steps needed, and attempts to illustrate the requirements and the opportunities given to allow each computer system to be shown at its honest best.

The overall VECTOR process is outlined in Figure 1. When thc user wishes to consider a new application, the procedure is entered at the point labeled "New Problem." These relativcly time-consuming series of steps need to be executed only once, to define and systematize the relevant characteristics of each new problem that is to bc considered.

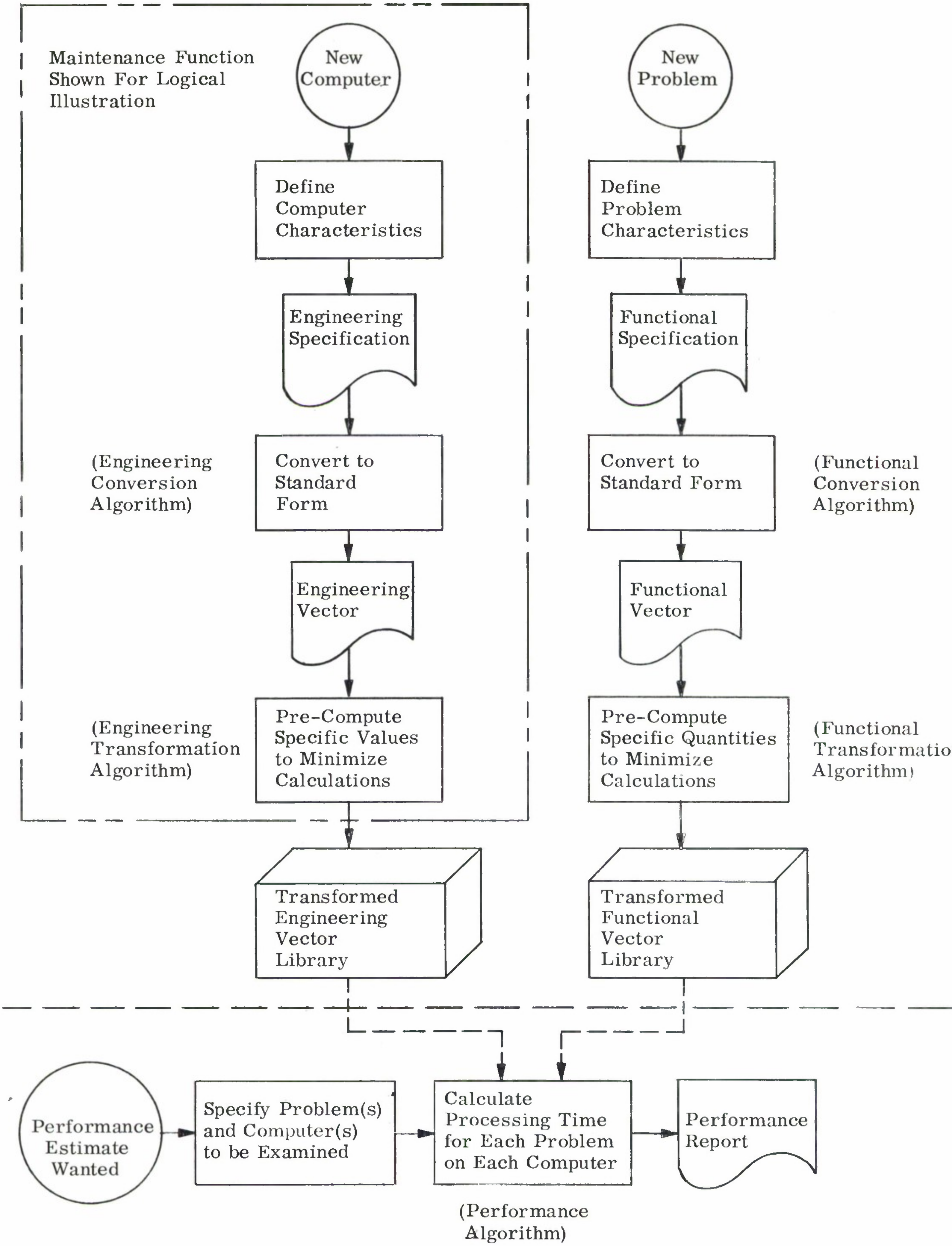


Figure 1. Schematic of the Vector Estimating System

Then, when the user wishes to determine the performance of a specific, previously-defined computer on a specific, previously-defined problem, he has only to enter at the point labeled "Performance Estimate Wanted," draw the relevant computer and problem characteristics from their respective libraries, and perform simple arithmetic calculations to produce the estimated total processing time.

The current version of the VECTOR process is limited to file processing applications on computer systems using magnetic tape for all on-line input and output, and all calculations are performed by hand. The method can be extended to handle many other classes of problems and computer configurations, and most of the procedural steps can readily be computerized.

1.2.2 General Description of the VECTOR Process

As part of the research program to develop the "VECTOR" process for quantitative evaluation of the information processing capabilities of general-purpose digital computers, AUERBACH Corporation has developed the following:

(1) A detailed set of Engineering Specifications which quantitatively describe the computer's basic characteristics. These numeric quantities represent the measures of computer time necessary to achieve a variety of standard tasks to be found, in varying degrees, in all computer programs. The tasks include data-conversion loop times, table-reference task time, character-editing loop times, as well as such basic items as the time required to add one eight-digit operand to another and store the result. Additional time measurements are required for effective tape speeds and central processor "overheads" or input-output penalties imposed by the transfer of data into or out of the main storage area.

The Engineering Specifications will permit any computer manufacturer to specify time measurements for any given computer system configuration.

(2) A set of Engineering Vectors, the elements of which describe the computer in terms suitable for performance evaluation.

(3) An Algorithm (precise procedure) for deriving Engineering Vectors from the Engineering Specification (Algorithm 2). The algorithm is divided into two parts: an Engineering Conversion Algorithm which converts the raw Engineering Specifications into the rigidly standardized Engineering Vector; and an Engineering Transformation Algorithm which transforms the pure vector into a form designed to minimize the calculations involved in estimating the performance of the computer upon a series of application tasks.

(4) A detailed set of Functional Specifications which quantitatively describe a basic data processing task. These numeric values represent statements of file volumes, record sizes, field lengths, input and output format requirements (if any), and any knowledge of the degree of complexity necessary to accomplish the processing required by the application demands.

The Functional Specifications will permit a prospective user to state the quantitative values relative to the parameters of a given application.

(5) The Functional Vector, the elements of which describe the application in terms suitable for evaluation of a computer's performance on the application task.

(6) An Algorithm for deriving the Functional Vector from the Functional Specification (Algorithm 1). The algorithm is divided into two parts: a Functional Conversion Algorithm which converts the raw Functional Specifications into the rigidly standardized Functional Vector; and a Functional Transformation Algorithm which transforms the pure vector into a form designed to minimize the calculations required to estimate the performance of a series of computers upon the application task.

(7) An Algorithm for estimating the time for a specified computer to perform a given application task (Algorithm 4). The inputs are the appropriate Functional and Engineering Vectors. The procedure is dynamically variable according to the contents of both the Functional and Engineering Vectors. For each variation in elements included or excluded, modifications must be added to or deleted from the base procedure.

1.2.3 Conclusions

Our research into the evaluation of computer system performance has produced the following conclusions:

(1) There is no magical formula approach for estimating computer system performance. However, reasonably accurate estimates can be produced at a rate of approximately one per hour after an investment outlay of one technical man-week per computer system and one technical man-day per problem.

(2) The accuracy desired in the final timing estimate is quite sensitive to the depth of programming detail applied to the measurement and the contents of the specifications and vectors. Most of the work to date has been based on providing outputs which are accurate to ±20% (i.e., in relation to results that could be obtained if the application task were actually programmed and run on the computer).

(3) There are a number of interesting interactions and sensitivities between the functional elements and the engineering elements and within the engineering elements. For example;

- The degree of control the system designer has over input/output formats may significantly affect operating times. Specifically editing operands and print formats requirements, input data and code conversion operands, tape format arrangements and operand sizes.

- The performance figures contained within the Engineering Vector must vary depending upon whether the process is input/output, storage or central processor limited, or is not limited by any of these conditions. To allow for these variations while maintaining sensitivity, we developed the necessary details for each condition separately. A solution to this is to develop a procedure within Algorithm 4 which examines all the major limiting cases and selects the smallest time limiting situation. (This factor can make significant differences in timing well beyond ±20%.)

- It has been determined that it is more economical of time and effort to bring some of the computation of the four cases into Algorithm 2 to reduce overall evaluation effort. Thus, many Engineering Vectors are given for four cases. For example, there can be separate estimates for "table look-up time" for the four cases: general, input/output limited, storage limited, and central processor limited.

(4) The present thinking and documentation as detailed in the VECTOR process presents, to the best of our knowledge, the most economical way of preparing accurate computer system performance estimates. It is our feeling that the preparation of the Transformed Engineering Vector, through the completion of the Engineering Specifications and the Engineering Conversion Algorithm and Engineering Transformation Algorithm should take an experienced analyst less than a man-week. Completion of the similar evolutionary process for derivation of the Transformed Functional Vector is only a matter of hours for an experienced analyst. Actual combination of the two vectors in the Performance Algorithm[1] to produce a time estimate, requires less than an hour.

(5) A valuable by-product of the VECTOR process is an indication to the prospective user of potential problem areas for each specific computer in preparation of the key program runs.

(6) The systems analyst, by following the standard procedure used to describe the application parameters in the VECTOR process gains a better appreciation of the important parameters to be gathered in a preliminary systems analysis phase.

(7) Judging from the increased accuracy and wider scope built into the VECTOR process in the course of its evolution, we feel that the potential of the quantitative measurement of computer systems can be further exploited by use of a general-purpose digital computer program to provide examination of the effect of parameter variations on the performance estimate.

---

[1] See Appendix III.

1.3 APPROACH

The objectives of the research project included the desire to develop estimating procedures that could produce performance estimates on any or all available computer systems for a given set of parameters that would specify an application task. The procedure was also designed to use only a minimum amount of technical time and effort.

In order to meet these aims and to produce performance estimates for computer systems on any given application, it was initially decided to evaluate key runs rather than an entire application series. Secondly, in order to make possible rapid production of timing estimates for any set of application measures provided in standardized form, (known as the Functional Vector), it would be desirable to pre-calculate the standard measures of performance in the form of the Engineering Vector. By establishing a library of Engineering Vectors (computer performance parameters) duplication of effort is eliminated and production of an accurate estimate is available in a matter of hours.

With these conflicting objectives in mind, and our own desire to make this procedure, now known as the VECTOR process, capable of producing a projection consistent with a preliminary systems analysis, a decision was reached midway in the research project that the number of Engineering and Functional Specifications would have to be expanded from a simple set originally envisioned and documented into Technical Note 1, May 29, 1963. We recognize that in many cases, the more detailed stipulations of sets of specifications for both Engineering and Functional Vectors would not be completed for each and every computer system, and/or functional application, but that their absence might introduce serious errors in the performance estimates. This argument can be stated; each computer system requires an analysis of the problem which is peculiar to its performance characteristics, and if it is necessary to know what the critical constraint is in each performance, then, since there are only a few possible critical constraints in any computer system, it is possible to do a separate analysis of the performance on each critical path. From this, the user of the VECTOR process can form solutions for each programming approach (i.e., storage limited, central processor limited, etc.). Also, as a by-produce of the VECTOR process the user will know the most desirable approach to structuring the program and its application processes. This flexible approach was felt to be more practical than any attempt to develop "absolute" performance which would still be complex and non-standard in content.

In the original concept of the Engineering Vector development, it was expected to carry all of the Engineering Specifications (i. e., tape times, tape packing density, time to add to operands and store the result, time to make a single loop of a binary search subroutine, etc.) into Algorithm 4 (Performance Algorithm) in which the performance was evaluated for each problem. Once we adapted the concept of providing four different performance figures for each Engineering Specification and resultant for strings of elements to produce four Transformed Engineering Vectors, it was felt that a great deal of time and effort could be saved if these performance figures, at the intermediate level, were precomputed. Coincidental with this, it became necessary to rewrite the entire concept of the Engineering Algorithm into the two parts previously described.

After the second version of the VECTOR process had been developed, a series of tests designed to test the practicability of the procedure was performed. These consisted of selecting programmers with diverse background to prepare and use the Transformed Engineering Vector for the three computers selected as prototypes - namely the UNIVAC III, IBM 7074, and Honeywell 800. No attempt was made at this preliminary stage to evaluate the results. Our only conclusion at this stage was that the process, in spite of its apparent volume, could be easily completed in a matter of days by experienced analysts. It appears that one computer analyst man-week should suffice to establish and process the computer description; one system analyst man-day should be sufficient to convert the application specifications to the Transformed Functional Vector; and one technical assistant man-hour should be sufficient to put the two together to produce a performance estimate. Naturally, when the computer analyst or the systems analyst have completed their task, it does not have to be repeated for subsequent cases.

Subsequently, the problem became one of limiting the scope of the current version of the VECTOR model so that it would fit within the limitations of our original objectives — namely time and level of complexity. In presenting our current version, we are aware that additional accuracy could be built into the process by adding more flexible measurements of computer performance. However, when these are weighed against increasing the complexity of the model, the decision must be made upon how much increased accuracy can be gained for the additional preparation work involved.

The current version of VECTOR process has been subjected to testing against AUERBACH Standard EDP Reports and the results are described in Paragraph 2.5.

SECTION 2. TECHNICAL DETAILS

2.1 MODELS

Three major model concepts can be distinguished in summarizing the progress of the research. Sample pages from each, flow charts, etc., are included in this section.

The original (Model A) was envisioned as using precise, unambiguous data from the Engineering Specifications to directly establish performance timings. In testing it worked, provided that the facts queried were really unambiguous; but this condition was not universally obtainable. Such items as effective tape speed, time to edit a character, etc., were found to depend on a number of factors. If each of these factors was given an arbitrary or average value, then the resulting Vector element was grossly inaccurate.

A second model was then constructed which replaced some of the major elements by establishing conditional performance data, instead of giving flat figures. This required that the operating environment of the problem was established before the associated performance data could be selected. In test, this gave much better figures and the logic of the procedure was carried over into the third model.

This final model differed from the second mainly in arrangement. There were two reasons for this, both of which are not necessarily compatible. One was to reduce as far as possible the amount of work involved in handling the process. The second was to make the whole rationale as easy to understand and appreciate as was possible. This latter was considered an essential part of the project since we felt that in order for the Vector process to be gainfully used, it would have to be readily understood by prospective users.

In the following paragraphs, brief descriptions of each model and of various Vector elements will be given, together with the precise reasoning as to why the various changes which were introduced into succeeding models were considered necessary.

2.1.1 The First Model

A list of the Functional Characteristics that would describe the problem, and a parallel list of Computer or Engineering Characteristics stating the parameters of the computer chosen, were processed to make them parallel in form, and used to estimate a timing. Samples of the lists are shown in Figures 3, 4, 5 and 6. A general flowchart of this Model is shown in Figure 2.

Typically, there were two types of problems involved in these timing estimates. One was the straightforward performance times of the magnetic tape units, such as developing effective performance times from some half-a-dozen magnetic tape unit characteristics, and from a character count of the records.

The second problem area seemingly capable of being precisely measured, raised many sided questions of what could be placed in the Engineering Vector relative to such concepts as Unpacking, Strict and Loose format, and other data dependent problems. These were defined independently of the data description in the first place, with a provision for later correction (tuning) designed to improve the accuracy. The tuning procedure allowed a particular Engineering Characteristic to be given in a number of forms, only one of which need be unqualified. The other cases were allowed to be conditional upon values elsewhere in the process, including performance times. Thus, a magnetic tape performance might be given as 30,000 characters per second; or if central processor timing is less than input-output timing, as 40,000 characters per second.

This would cause the 30,000 characters per second figure to be used in the initial determination of the performance estimate, and then, if these timings showed the whole to be input-output limited, the performance would be recalculated using the higher magnetic tape rate.

The aim of this was to avoid either missing opportunities or falling into traps by using figures which were only valid in some specific circumstances.

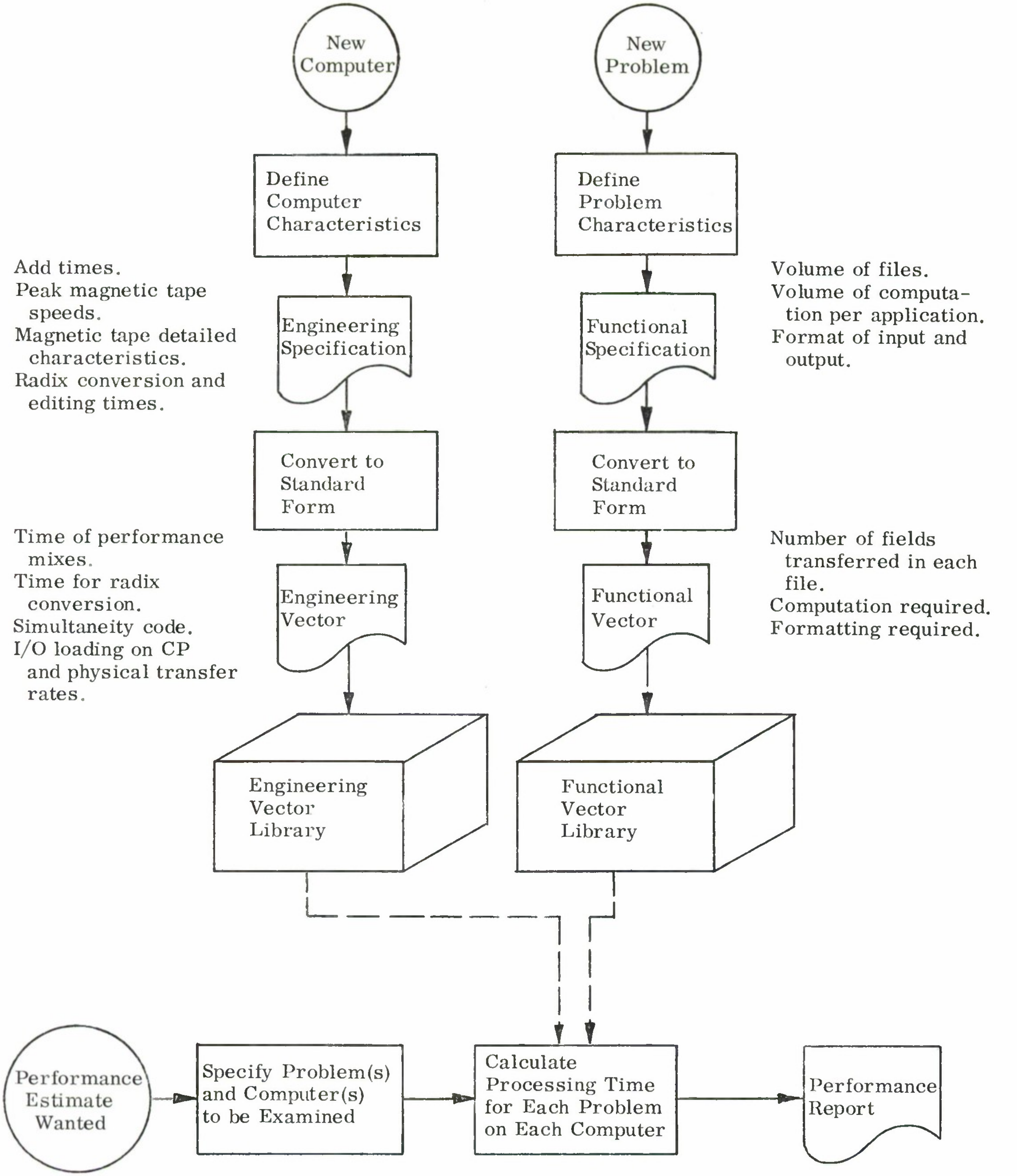


Figure 2. Flowchart of the First Model for the Vector Process Showing Examples of the Specification and Vectors, Based on Physical Descriptions Only.

ES NUMBER	ENGINEERING SPECIFICATION	UNIT(S) OR CODE REFERENCE OF ENGINEERING ITEM
	Central Processor	
1000	Standard size of computational module	Code Category from ES1; μsec.
1010	$c = a + b$	Code Category from ES1; μsec.
1020	$b = a + b$	Code Category from ES1; μsec.
1030	Sum N items	Code Category from ES1; μsec.
1040	$c = a * b$	Code Category from ES1; μsec.
1050	$e = a/b$	Code Category from ES1; μsec.
1060	$c_i = a_i + b_j$	Code Category from ES1; μsec.
1070	$b_j = a_i + b_j$	Code Category from ES1; μsec.
1080	Sum N_i	Code Category from ES1; μsec.
1090	$c = e + a_i b_j$	Code Category from ES1; μsec.
1100	Branch based on comparison of (numeric) magnitude	Code Category from ES1; μsec.
1110	Branch based on comparison of (alphameric) magnitude	Code Category from ES1; μsec.
1120	Branch dependent on unchecked data value	Code Category from ES1; μsec.
1130	Branch dependent on checked data value	Code Category from ES1; μsec.
1140	List Search	Code Category from ES1; μsec.
1150	Format Control/Char. (Unpack)	Code Category from ES1; μsec.
1160	Format Control/Char. (Pack)	Code Category from ES1; μsec.
1170	Table look-up per comparison	
	For a match	Code Category from ES1; μsec.
	For least or greatest	Code Category from ES1; μsec.
	For interpolation point	Code Category from ES1; μsec.

Figure 3. Extract from the Engineering Specification of the 1st Model for the VECTOR Process Showing Specific Timings Being Requested, Without Regard to Circumstances.

2. FUNCTIONAL SPECIFICATION

2.1 DRAFT LIST OF FUNCTIONAL SPECIFICATIONS (Including Code Category Definitions)

FS NUMBER	FUNCTIONAL SPECIFICATION	UNIT(S) OR CODE REFERENCE OF FUNCTIONAL ITEM
0010	Number of Main File Records	Number.
0020	Record Size, Main File	Code Category from FS1 with 1, 2, or 3 numbers.
0030	Record Type, Main File	Code Category from FS2.
0100	Transaction Activity Volume - Average	Number of items per cycle.
0110	Transaction Activity Volume - Peak	Number of items per cycle.
0120	Transaction Activity Distribution	Code Category from FS5.
0210	Record Size, Transaction	Code Category from FS1 with 1 or 2 numbers, plus number of transactions involved. It is possible to have more than one transaction record size per application.
0220	Transaction Media	Code Category from FS4.
0310	Record Size, Reports	Code Category from FS1 with 1 or 2 numbers, plus the number of transactions involved.
0320	Reporting Media	Code Category from FS4.
0330	Reporting Format	Code Category from FS6.
0340	Computation Type	Code Category from FS3.
0350	Computation Index	Code Category from FS7.
0410	Application Cycle Time	Days, Hours, Minutes.
0420	Application Percentage of Installation Utilization	Percentage.
0430	Hourly Utilization Expected Each Week	0 through 168.

Figure 4. Extract from Functional Specification of the 1st Model for the VECTOR Process Showing Computation Being Requested per Application.

DRAFT LIST OF ENGINEERING CHARACTERISTICS
(Including Code Category Definitions)

EC NUMBER	ENGINEERING CHARACTERISTICS	UNIT(S) OR CODE REFERENCE OF ENGINEERING ELEMENTS
0010	Computational-Mix Indices	Encoded using EC1.
0020	Conversion and Translation Indices for each class of peripheral	6 Elements encoded using EC2.
0030	Configuration Types and Properties for each class of peripheral	5 Elements, n entries, using EC3. Note: Load includes verification and transfer only.
0040	Configuration Simultaneity	1 Element from ES 1500.
0050	Configuration Cost	9 Elements ES 2010 - ES 2040.

Figure 5. Extract from Engineering VECTOR of the 1st Model for the VECTOR Process

DRAFT LIST OF FUNCTIONAL CHARACTERISTICS
(Including Code Category Definitions)

FC NUMBER	FUNCTIONAL CHARACTERISTICS	UNIT(S) OR CODE REFERENCE OF FUNCTIONAL ELEMENTS
0010	File Description(s)	n elements per file, encoded as FC No. 1, 2.
0020	Computation Index	Encoded as FC No. 2.
0030	Application Description	Encoded as FC No. 3.

Figure 6. Extract from Functional VECTOR of the 1st Model for the VECTOR Process.

2.1.2 The Second Model

During the testing of the first model, two points became apparent.

(1) The details being asked for were inadequately defined, calling as they did for "general" (average) performance figures. Different analysts honestly disagreed on what they were timing and no uniformity seemed possible even with redefinition.

(2) A number of calculations were repeated over and over again, in fact were repeated whenever a particular set of characteristics were present.

These difficulties were resolved by defining four standard "Performance" conditions (such as when the application is processor limited) and gathering for each and every characteristic a separate performance measure for each situation. The circumstances under which each performance was to be obtained was now well-defined and unambiguous, and the fact that the various performance figures were themselves carried in the vector eliminated much of the repetitive work.

The resultant model is shown in Figure 7. Here the application specifications remain as they were in the first mode; and so do the more precise measures of the engineering model. The less firm ones (specifically those dealing with editing concepts, radix conversion times, etc.) were now defined directly within specific categories.

Four standard "Performance" factor categories were defined:

(1) When the application was known to be central processor limited by at least 20%.

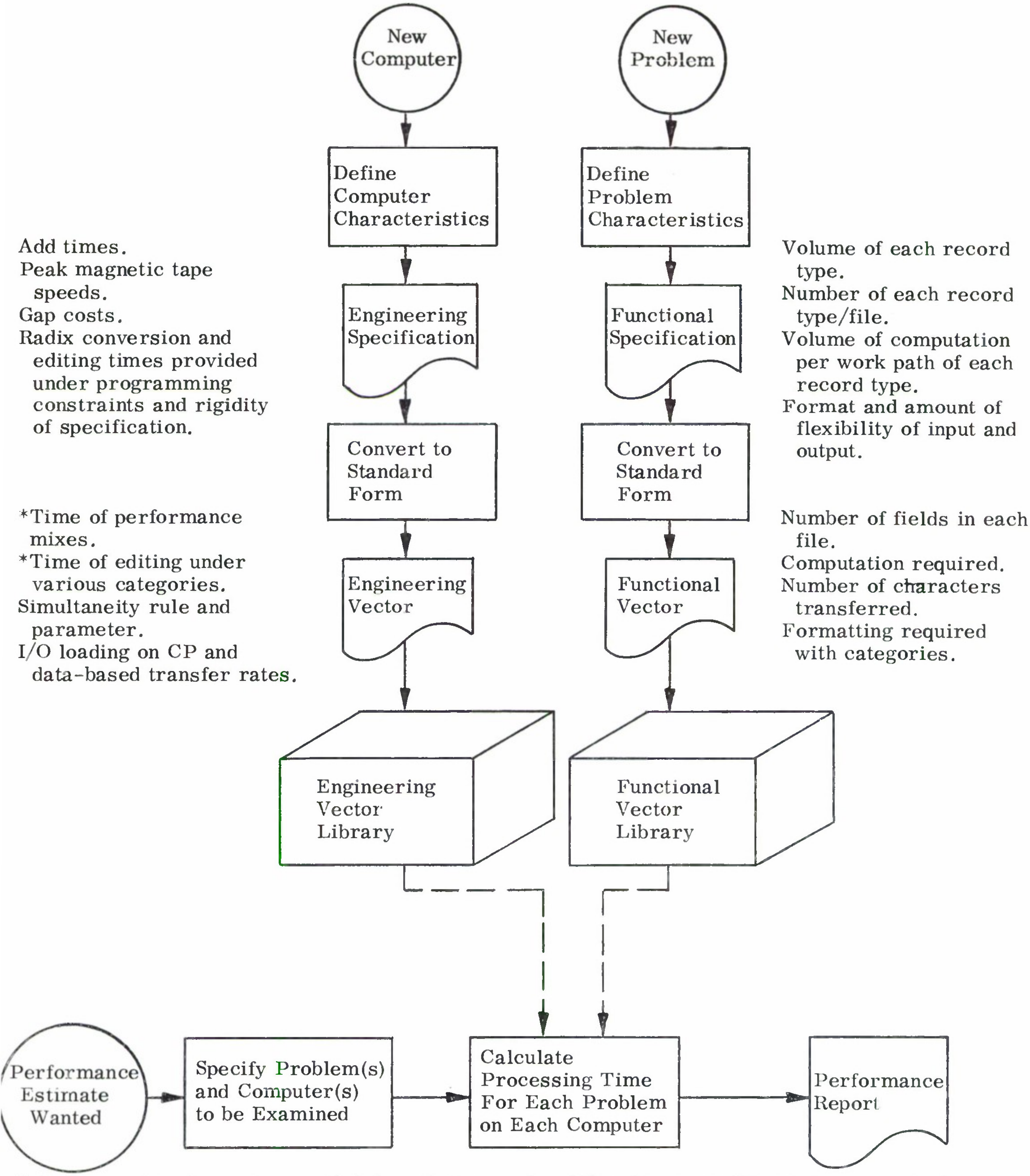
(2) When the application was known to be I/O limited by at least 20%.

(3) When the core storage space was known to be short.

(4) When nothing was known about the limiting factors.

The Engineering Specifications were then to be processed into estimated performance figures, one for each performance factor and the Vector elements were changed accordingly (compare the Vector elements in Models 1 and 2 in Figures 2 and 7).

The precedure employed in the second version was to use the general (Category 4) case in the first pass through the Engineering Algorithm and then, noting what conclusion it produced regarding the final timing and what the limiting factor was, the Performance Algorithm was then repeated, using the elements of the Vector indicated by the limiting factor. In order to achieve stability (i.e., a performance time based on a critical factor that is minimized but which remains the limiting factor,) it was found that it might be necessary to work through the Performance Algorithm still another time with another set of Vector elements. This would establish how close the results would be. It also indicated a most valuable by product of the entire VECTOR process; that is, an indication of what the limiting factors affecting program running time would be and what could be done about them. Specimen pages are shown in Figures 8 to 13.



* Indicates this figure is provided four times, each valid under a specific program constraint.

Figure 7. Flowchart of the Second Model for the Vector Process Showing Examples of the Specification and Vectors, Based on the Performance Under Specific Constraints.

SPECIFICATION 1

QUERY	ANSWER	COMMENT
Time taken to add an 8-digit operand to another, both being in main store. ES101		
Time taken to multiply an 8-digit operand by an n-digit one. ES102		
Time taken to divide an 8-digit operand by an n-digit one. ES103		
Size of word in AN chars. ES104		
Size of word in numeric chars. ES105		
No. of index registers. ES106		
No. of words in main store. ES107		
No. of words that can be reached without indexing. ES108		
Time to index an operand. ES109		
Time to indirectly address an operand. ES110		
Percentage of store which can be reached by indexing from an arbitrary position. ES111		
Code and time the comparison of two 8-digit numeric fields held in storage, and the execution of a jump to an arbitrary location based on the comparison. ES112		
What is the time used to enter one of six independent routines based on a digit (between one and six) held elsewhere in storage? ES113		

Figure 8. Extract from the Engineering Specification of the 2nd Model for the Vector Process Showing Specific Timings Being Requested Where Circumstances are Suitable.

SPECIFICATION 1 (Cont'd)

QUERY	ANSWER	COMMENT
What is the time involved in placing single digit in the center of an otherwise unaltered word in main storage. The digit is assumed to be right justified in the accumulator or in another storage position. ES116		
What is the time involved, when using the standard routines, (if any) to take a 5-char numeric field in punched card code lying within a tape input area, but not aligned to the word structure of the system, and place it, unpacked, into a word or words ready for use as an operand. ES201		
What is the time involved, when using the standard routines, (if any) to take an 11-character alphabetic field in punched card code and lying within a tape input area but not aligned to the word structure of the system, and place it, unpacked, into a word ready for a comparison operation. ES202		
What is the time involved, when using the standard routines, (if any) to take a 5-char numeric field in punched card code lying within a tape input area, and aligned to the word structure of the system, and place it, unpacked, into a word or words ready for use as an operand. ES203		

Figure 9. Extract from Engineering Specification of the 2nd Model for the VECTOR Process Showing Times Being Requested Under Specific Constraints Where Appropriate.

PART 1

This part deals with the content and form of the files to be processed. One type of Main File Record; Two Types of Transaction Files, and Two Types of Report Files can be described here. If more are needed, describe these in continuation forms.

QUERY	ANSWER	ASMT.	CN
Main File Details			
No. of Records		-	
No. of Chars/record		-	
No. of known numeric characters/record		50%	
Anticipated storage media		M Tape	
Transaction File Type 1			
No. of Records		-	
No. of Chars/record		-	
No. of known numeric chars/record		50%	
Anticipated storage media		Card	
Will it be sorted in order of main file ?		Yes	
May the analyst change the format of the records ?		Yes	
Transaction File Type 2			
No. of Records		-	
No. of Chars/record		-	
No. of known numeric chars/record		50%	
Anticipated storage media		Card	
Will it be sorted in order of main file ?		Yes	
May the analyst change the format of the records ?		Yes	
Report File Type 1			
No. of Records		-	
No. of printed lines/record		1	
Anticipated storage media		Paper	
Should it be in main file order ?		Yes	
Document length in inches		Yes	
No. of computed chars per line		60	
May the analyst change the format of the report ?		Yes	
Report File Type 2			
No. of Records		-	
No. of printed lines/record		-	
Anticipated storage media		Paper	

Figure 10. Extract from Functional Specification of the 2nd Model for the VECTOR Process Showing Breakdown by Record Type.

GOVERNING CONDITION ______________

EV	COMPONENT	VALUE
1.	Time taken to edit an arbitrary numeric field during input.	μsec.
2.	Time taken to edit an arbitrary alphameric field during input.	μsec.
3.	Time taken to edit a flexible numeric field during input.	μsec.
4.	Time taken to edit a flexible alphameric field during input	μsec.
5.	Time taken to complete a Simple Update operation.	μsec.
6.	Time taken to complete a Complex Update operation.	μsec.
7.	Time taken to complete a Table Reference operation.	μsec.
8.	Time taken to edit an arbitrary numeric field during output.	μsec.
9.	Time taken to edit an arbitrary alphameric field during output.	μsec.
10.	Time taken to edit a flexible numeric field during output.	μsec.
11.	Time taken to edit a flexible alphameric field during output.	μsec.
12.	Magnetic Tape Unit Performance in AN characters, corrected by AN data characters/character position efficiency ratio.	char/msec.
13.	Magnetic Tape Unit loading on the central processor per AN data character.	μsec/char.
14.	Magnetic Tape Unit Performance in decimal characters, corrected by decimal data character/character position efficiency ratio.	char/msec.
15.	Magnetic Tape Unit loading on the central processor per decimal data character.	μsec/char.
16.	Magnetic Tape Unit Performance in binary characters, corrected by binary data character/character position efficiency ratio.	bits/msec.
17.	Magnetic Tape Unit loading on the central processor per binary data character.	μsec/bit.
18.	Magnetic Tape Unit rate for the input of card images.	cards/msec.
19.	Magnetic Tape Unit loading on the central processor, per card-image input.	μsec/card

Figure 11. Extract From Engineering Vector of the 2nd Model for the Vector Process Showing Precise Names Used For Values Which By Nature Are Not Precise, Thus Obstructing Easy Understanding.

COMPONENT	VALUE
Number of numeric fields requiring strict editing on input.	
Number of alphameric fields requiring strict editing on input.	
Number of numeric fields requiring a flexible editing on input.	
Number of alphameric fields requiring a flexible editing on input.	
Number of times a simple updating process is executed.	
Number of times a complex updating process is executed.	
Number of times a table reference is executed.	
Number of numeric fields requiring a strict editing for output.	
Number of alphameric fields requiring a strict editing for output.	
Number of numeric fields requiring a flexible editing for output.	
Number of alphameric fields requiring a flexible editing for output.	
Total characters in the Master File	
Alphameric characters in the Master File.	
Numeric characters in the Master file.	
Total characters in the Master Record.	
Alphameric characters in Master Record.	
Numeric characters in Master Record.	
Total characters in Transaction File.	
Storage Form of Transaction File.	
Number of lines in Report File.	
Number of Master File Records.	
Number of Transaction File Records.	

Figure 12. Extract from Functional VECTOR of the 2nd Model for the VECTOR Process Showing Elements of the Vector.

	A	B	C	D	E	F
TOTAL CHARS. MASTER FILE (1)		MTU PERFORMANCE (AN) (char/msec)		$\frac{(2A)}{(1C)} + \frac{(3A)}{(3C)} \rightarrow 1E$		MAIN FILE ELAPSED TIME (DECIMAL)
ALPHA CHARS. MASTER FILE (2)		COST (AN) (μsec/char)		$\frac{(2A)}{(1C)} + \frac{(3A)}{(5C)} \rightarrow 2E$		MAIN FILE ELAPSED TIME (BINARY)
NUMERIC CHARS. MASTER FILE (3)		MTU PERFORMANCE (DECIMAL) (char/msec)		$(2A) \times (2C) + (3A) \times (4C) \longrightarrow 5E$		TRANSACTION FILE ELAPSED TIME
(4)		MTU COST (DECIMAL) (μsec/char)				REPORT FILE ELAPSED TIME
ALPHA CHARS. MASTER RECORD (5)		MTU PERFORMANCE (BINARY) (char/msec)		$\frac{(7A)}{(7C)} \longrightarrow 3E$		CENTRAL PROCESSOR USAGE (DECIMAL)
NUMERIC CHARS. MASTER RECORD (6)		MTU COST (BINARY) (μsec/char)		$(7A) \times (8C) + (5E) \longrightarrow 5E$		CENTRAL PROCESSOR USAGE (BINARY)
TOTAL CARD IMAGES TRANSACTION FILE (7)		MTU RATE FOR CARD IMAGE INPUT (CARDS/msec)		$\frac{(9A)}{(9C)} \rightarrow 4E$		

Figure 13. Extract from Computational Algorithm of the 2nd Model for the VECTOR Process Showing Varying, More or Less Complex, Procedures.

2.1.3 The Third Model

The results of the second model appeared technically promising; but the format adopted, while usable, was not as clear as it could be. In addition, confusion arose when technical agreement was sought on development of the values for the Vector elements. In the second model, these had been put in terms of performances for each of the items within the problem specification; and as such they were very much subjective quantities. This proved to produce unnecessary complications in the minds of computer experts, accustomed to dealing in precise, unquestioned data.

In addition, formats had been designed for two conflicting circumstanccs — to minimize the computational tasks and to show the rationale of the process.

A number of experiments were made in formats (See Figure 14, 15 and 16) for computational and explanatory use. The most satisfactory proposal brought forth was to introduce a new entity, a Transformed Vector into the procedure. (See Figure 17). This would be a logical derivation from the actual Vector, but the transformation would include all the assumptions and compromises built into the model. Thus, the Vcctor quantities would be precise and as such acceptable, yet the computation would be as simple as possible.

Following this, it was agreed that it was most necessary at this time for the system to be clearly understood. The final formats utilized, particularly those included in the transformations, spelled out the processes and assumptions involved (see dctail in Figure 18). At the same time, it was possible to simplify the estimation of a specific time elemcnt so that it would consist solely of a number of cumulative multiplications. (Figure 19).

COLUMN	1	2	3	4	5	6
EV24R	=1	=1	=2	=2	=3	=3
EV24P	=1	≥ 2	=2	≥ 4	=1	≥ 5
PV6	Max [(PV2+PV3), (PV2+PV4)]	Max [(PV2, PV3, PV4)]	Min { Max [2(PV2) + (PV3), (PV4)]; Max [2(PV2) + (PV4), (PV3)]; Max [(PV2 + PV3), (PV2 + PV4)]; Max [2(PV2), (PV3 + PV4)] }	Same as Column 2	Min {Max (PV1, PV1 + PV 2+ PV3 + PV4); etc.	Same as Column 2

Figure 14. Extract from Computational Algorithm of the 3rd Model for the VECTOR Process Showing Experimental Format.

GIR 1		
PAGE 1A		
Specification	Symbol	Formula
G, T, S Edit Input Times Card 1	(IR1) =	5N + 4A
(AN party, non-aligned)	N =	$(ESE6) - [(ESE6) - (ESE2)] [\min (1, \frac{3}{(ES105)})]$
	A =	$(ESE11) - [(ESE11) - (ESE9)] [\min (1, \frac{1}{(ES104)})]$
Card 2 (AN aligned)	(IR2) =	5(ESE2) + 4(ESE9)
Card 3 (numeric party non-aligned)	(IR3) =	9N
Card 4 (numeric aligned)	(IR4) =	9(ESE2)

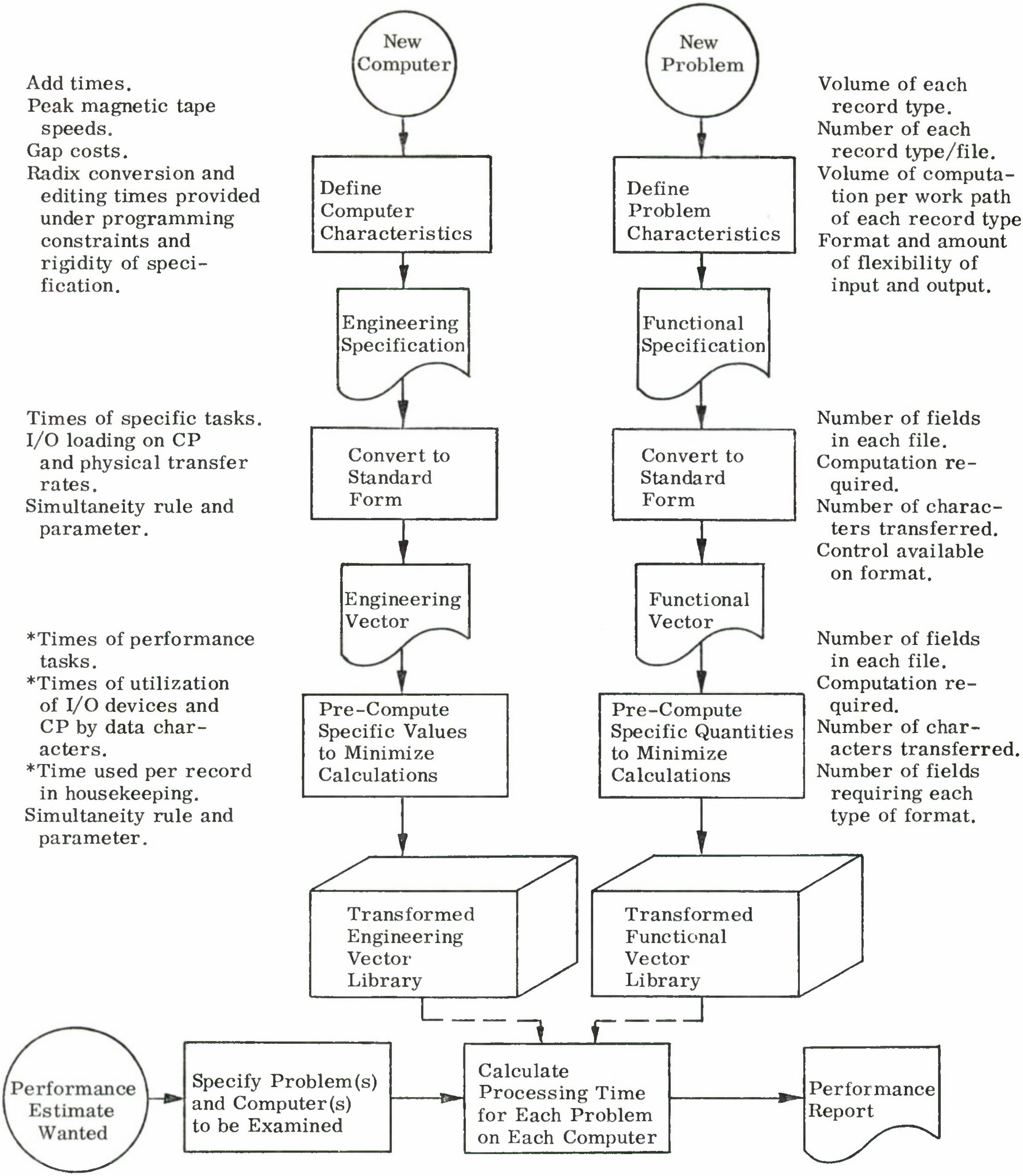
NOTE: If ESE given as a formula in n = no. of char per field, use n = 5 for ESE2, ESE6; n = 11 for ESE9, ESE11

Figure 15. Extract from Definition of Engineering VECTOR of the 3rd Model for the VECTOR Process Showing Experimental Format in Equation Form.

OPERAND ADDED COSTS					
(ES109) > (ES110)		Yes I	No	No	No
(ES106) > 10		-	Yes II	No	No
(ES111) (ES107)(ES104) < 40,000		-	-	No	Yes III
		3E	1E	1E	2E
G	IR9	(ES110)	$(ES109) + \frac{(ES115)}{4}$		$2(ES109) + \frac{(ES115)}{4}$
T	IR10	(ES110)	$(ES109) + \frac{(ES115)}{4} + \frac{(ES114) + (ES115)}{2}$		$2(ES109) + \frac{(ES115)}{4} + \frac{(ES114) + (ES116)}{2}$
S	IR11	2(ES110)	(ES109) + (ES115)		2(ES109) + (ES115)
P	IR12	0	0		0

I. indirect addressing preferably to indexing.
II. enough index registers.
III. small store

Figure 16. Extract from Definition of Engineering VECTOR of the 3rd Model for the VECTOR Process Showing Experimental Format Using Mixed Decision Table and Equation Approach.



* Indicates this figure is provided four times, each valid under a specific program constraint.

Figure 17. Flowchart of the Third Model for the Vector Process Showing Examples of the Specification and Vectors, Based on Physical Descriptions and Performance under Specific Constraints, and Transformed Vectors to Simplify Computation.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON DECIMAL DATA under I-O Critical conditions.

Pre-requisites: PR 1901, PR 1902, PR 1918, PR 1919.

Method

This element can be arrived at by assuming that PR 1918 correctlv estimates the block size to be used and PR 1919 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation, the value of the element concerned is:

((PR 1901) x (PR 1919)) x ((PR 1918) ÷ ((PR 1918) + (PR 1902)))

= ((________) x (________)) x ((________) ÷ ((________) + (________)))

= ________________ digits per second.
(evaluated to 3 significant figures)

This gives a rate in digits/second. To transform this into the required microseconds/digit form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ________________ = ________________, which is TEV 19 (IO).

Figure 18. Extract from Transformation of the 3rd Model for the VECTOR Process Showing Assumptions Being Specified, and Working Space Provided.

PERFORMANCE ALGORITHM

Central Processor Times

TFV Description	TFV Value	TEV Description	TEV Value, μsec		RESULTS: (TFV Value) x (TEV Value)* General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 01 No. of fixed numeric input fields		TEV 01 Edit a fixed numeric field during input	G					
			CP					
			I-O					
			S					
TFV 02 No. of fixed alphameric input fields		TEV 02 Edit a fixed alphameric field during input	G					
			CP					
			I-O					
			S					
TFV 03 No. of flexible numeric input fields		TEV 03 Edit a flexible numeric field during input	G					
			CP					
			I-O					
			S					
TFV 04 No. of flexible alphameric input fields		TEV 04 Edit a flexible alphameric field during input	G					
			CP					
			I-O					
			S					
TFV 05 No. of Simple Update Operations		TEV 05 Simple Update Operation	G					
			CP					
			I-O					
			S					
TFV 06 No. of Complex Update Steps		TEV 06 Complex Update Step	G					
			CP					
			I-O					
			S					
Subtotals (to be carried to third page):								

*Round each result to the nearest second.

Figure 19. Extract from Performance Algorithm of the 3rd Model for the VECTOR Process

2.2 RATIONALE

There are a number of possible reasons for including a particular item in the specifications. It may be used to establish the volume of a file, or to check that the problem concerned is within stipulated boundaries or for various other purposes. Once the specification is available, however, it may be used in other ways, so that it is not always easy to establish what the primary purpose of a particular specification is simply by looking at the procedure and noting its use.

Accordingly, the reasons for the inclusions of various specifications is provided below, grouped under various categories within the specification concerned.

2.2.1 Engineering Specifications

There are eight basic reasons for an item being included in the Engineering Specification. These are:

(1) To describe the hardware organization of the computer.

(2) To describe the performance of the system as it is dictated by the design of the order code.

(3) To describe the performance of the system as related to the addition, comparison, and multiplication operations.

(4) To describe the performance of the system when receiving data from or transmitting it to the outside world.

(5) To describe the capacity of the system to provide for input/output operations.

(6) To describe the capacities and organization of each input/output device.

(7) To provide background data.

(1) Purpose: To describe the hardware organization of the computer

The memory size of a computer is clearly needed in such an estimating procedure, but in itself is insufficient. The useful memory will change with such things as whether it is a fixed or variable word; whether computation is in binary or decimal, and a number of other factors. Additional information to be supplied as comments should contain information on the use of overlapped core banks, "fast" or control memory and other special features of the hardware.

These and other factors therefore, need to be known before any estimating attempt can be made.

Specifications Pertaining to this Purpose

ES 201 through ES 211.

(2) Purpose: To describe the performance of the system as dictated by the design of the order code.

Typically an operational computer system of 1963 has an instruction repertoire of 10^2 instructions. However, this is not a necessary state, and there are commercially available systems with two or three times that number of available instructions. However, even here, the number of instructions that are actually employed to any significant extent is still limited to a selected few and this limitation appears to be continuing. Furthermore, many desirable instruction codes are not available in specific systems. Some, for instance, have no comparison instruction, only a jump on specific condition, while some have a three-way jump on a greater/lesser/equal relationship between two operands. Others have even more sophisticated relationships, particularly in table searching while still others do not offer multiply and divide instructions with the basic complement.

It is impractical to completely analyze all the different order codes since one would be hard put to establish valid criteria to fit all program demands. Equally the differences must not be glossed over. As a compromise, which appears satisfactory when viewed as part of the VECTOR process, some specific tasks are timed which are designed to bring out these differences; and then the performance which is obtainable on these tasks is used as a measure of the efficiency of the order code.

Specifications Pertaining to this Purpose

ES 306
ES 308 - 316
ES 450 - 453

(3) Purpose: To describe the performance of the system as related to the addition, comparison and multiplication operations.

Typically addition, and its related instructions, subtraction, and comparison, form an essential (or indeed the essential) core of the executed instructions within most tape oriented data processing problems. Equally typically, it is necessary to definc a specific operating task, as there are many individual ways in which an addition operation may be carried out depending on the system concerned. Therefore, each Engincering Specification is explained in detail as to what operations are to be performed. However, it is not the intent of the specification to indicate how the operation sequences should be performed.

Multiplication is the second computational ability of a computer system, and despite its close logical relationship with addition, is essentially independent of the addition timing for the purpose of performance. This occurs because again there are so many available ways of implementing a multiplication operation that the multiplication/addition time ratio is essentially a design decision which cannot be forecast.

The nature of the specifications are designed to make all computers perform the same tasks. It is only through consistent standardization of a variety of programming tasks that one can eliminate such charges as "my machine multiplies n times better than the competition but your analysis only looks at addition operations" or "use of eight digit operands only is unfair to a machine with a six digit fixed work length." Employment of both types of features helps to provide a means of eliminating these claims and counter claims.

Specifications Pertaining to this Purpose

ES 301 - 315
ES 307

(4) Purpose: To describe the performance of the system when receiving data from, or transmitting it to the outside world.

This is a specific part of the load on a computer system, and consists of the recognition, verification, translation as needed and editing of data when passing from and to the computer/outside world interface. It is an area which has been frequently neglected in making computer timing estimates yet the whole problem of data manipulation can be equally important in its contribution to the total time as the computational process.

Using the COBOL language specifications as a base, we become aware of the two major factors concerning data manipulation:

a.	Alphabetic data	vs	Numeric data
b.	Fields aligned to computer word structure	vs	Fields non-aligned to computer word structure.

Within these there are problems of radix conversion, code translation, input vs output, rigid vs loose format requirements, etc. Our specifications are designed to provide input bearing on all of these factors. The machine type FW - AN, VW - N, etc., will have some bearing on how these specifications are answered. Also, the instruction repertoire and input-output organization must be considered by the analyst before he proceeds to code the standard editing task.

Specifications Pertaining to this Purpose

ES 401 - 430

(5) Purpose: To describe the capacity of the system to provide for input/output operations.

This is a measure of the types of simultaneity which exists between the various magnetic tapes, input/output trunks, and the central processor itself. It is provided in the form of a number of questions that will allow a simple analysis to reveal if one of a number of common input-output techniques is in use without asking the specifier to understand the full range of input-output simultaneity techniques which could be employed in the full range of presently available computer systems.

Specifications Pertaining to the Purpose

ES 501 - 507

(6) Purpose: To describe the capacities and organization of each input-output device.

This section covers the capacities and limitations of the specific device and its organization, both within the device itself, and also as across the interface with the computer system.

Different forms will have to be used for each type of device. At present only the form for magnetic tape units has been developed for the VECTOR process.

Specifications Pertaining to this Purpose

ES 601 -612

(7) Purpose: To provide background data.

At the conclusion of a VECTOR analysis, the user will have a timing estimate for each computer system processed. In order to evaluate these estimates properly, he will require additional background about the system, its history, its available software packages and library routines, its component prices, etc. This requirement is fulfilled by including in the specification a special section for this data, and then carrying this forward to the Transformed Vector.

Specifications Pertaining to this Purpose

ES 100 through 199

2.2.2 Functional Specifications

There are five basic reasons for an item being included in the Functional Specification. These are:-

(1) To provide background data.

(2) To provide details of the files in the process.

(3) To provide details of the volume of processing involved.

(4) To provide details of the type of processing involved.

(5) To ensure that the problem is suitable for the VECTOR process.

(1) Purpose: To provide background data

At the conclusion of a VECTOR analysis, the user will have a timing estimate based on a specific problem description. A number of factors will, however, be relevant, describing the environment in which the application will be performed. These are, however, included as part of the Functional Specifications and carried forward (without conversion) to the Transformed Vector.

It is our aim to gather data that will provide some background on the context of the key run under examination. In determining to proceed on a key run basis rather than on an entire application series, it is our belief that a few runs contribute to 30 to 50% of the total utilization computer time. The remainder of the time is spent in sorting (where we believe formulas present the most effective means of preparing running time estimates) and peripheral media conversion. Therefore, in order to get a true picture of that part of the installation load which can be measured, it may be necessary to put several proposed key runs through the VECTOR process.

Specifications Pertaining to this Purpose
FS 102 - 108

(2) Purpose: To provide details of the files in the process

In order to prepare any timing estimates, details pertaining to File and Record characteristics are needed. Some of these details will request information on alphabetic vs numeric character content, operand sizes, table sizes to be referred to by the program, and the number of record types within a field (to allow for multiple transaction types, summary and detail reports, header and trailers, etc.).

At the same time, no precise file design is assumed, or anticipated, since this can be modified by the findings reported after the initial pass through the Performance Algorithm. For example, if the overall performance is found to be tape-limited, it may even be advisable to consider how the "packing" of the master and/or transaction files would affect the overall performance.

Specifications Pertaining to this Purpose

FS 201 - 206
FS 310 - 340
FS 401 - 402
FS 510 - 540
FS 601
FS 710 - 750

(3) Purpose: To provide details of the volume of processing involved

The questions pertaining to volume of processing involved have been organized to determine the processing requirements only for those parts which are determined by the application, since the transfer of data blocks and records within blocks is a function of the computer system housekeeping that is accounted for from the number of records and the time to transfer a record. The description of the processing activity has been split between the master and transaction files, and within these into each of the major record types. The concept of seeking similar information for varying parts of the processing will permit us to estimate workloads more accurately and enable the program to be more specifically designed. Data is sought at this level for the volume of processing involved in each relevant type of record.

Specifications Pertaining to this Purpose

FS 350 - 390
FS 550 - 590

(4) Purpose: To provide details of the type of processing involved

To have any value practically, the processing to be measured must not confuse addition and multiplication. For the purpose of the VECTOR process three types of processing have been recognized - those based on addition, those based on multiplication, and those based on table searching. While there are other types of processing operations not covered within these three categories, it is expected that they can be indicated as some equivalent within one of the categories. Also, this use of only three categories is designed to make the analyst's job somewhat simpler since he can think in terms of threc operational categories - Simple, Complex, and Data Retrieval.

Specifications Pertaining to this Purpose

FS 360 - 390
FS 560 - 590
FS 320, 330, 340
FS 520, 530, 560
FS 720, 730, 740

(5) Purpose: To ensure that the problem is suitable for the VECTOR process.

In its present state many problems are not suitable for the process. A number of "boundary" questions have therefore, been asked so as to ensure that no misleading results are given. These include checking as to whether the problem is considered suitable for magnetic tape processing, or otherwise, a mass storage solution could be anticipated in some situations, if the transaction files are to be assumed to be on magnetic tape, whether the number of significant digits needed on each work path is within bounds, etc.

Should the answers to any of these queries not be within given bounds, then a user will be recommended not to use the VECTOR process without careful consideration as to the usefulness of the result.

Specifications Pertaining to this Purpose

FS 101
FS 103
FS 202
FS 204
FS 380
FS 580

2.3 ACCURACY

The degree of accuracy achieved in any estimating process can most readily be found by comparing the estimate with the actual result. However, the accuracy of computer system performance estimates cannot be practically measured in this manner. It would necessarily involve a large, rigorously controlled programming effort which would have to be repeated for each new computer system. The cost of such an effort would be out of proportion to the value of the results.

Another factor is also pertinent. Suppose such an effort were to be undertaken, and a program were to be written by expert programmers, guided by expert systems analysts, and then timed to form the basis of an accuracy evaluation. The question would then arise as to what elements to put into the program to guarantee an accurate measurement.

It is rarely possible to say exactly what an application task consists of, because the terms of the description itself will either be so detailed as to cover all the possible situations which may affect system design; or will be stated so vaguely that only estimates of relative sizes, contents, and volumes are possible. Each time a relevant detail is omitted, then the accuracy of the estimated timing on that specific computer system is reduced and less valuable. On the other hand, the moment a specific detail is included in the description, it becomes suspected of being oriented toward the solution of the problem on some specific computer system.

Thus, for instance, if no data on operand lengths is given, the estimates become less accurate on many systems; if some data is given in the form of a statement, such as "No operand field is greater than 8-digits," some systems (e.g., fixed word length systems) can be accurately timed for varied internal operations without any more information relative to the functional data. However, many of the timings for other types of computers (e.g., variable word length, or less than 8-digit fixed word length) could be badly biased. The only alternative would be to state that there are so many 1-digit operands, so many 2-digit operands, etc. This, however, would also be incorrect, as the stated operand lengths might well be based on the assumption that specific operations would, for instance, be performed using division rather than multiplication. Division by 4, for example, can be accomplished by division by a 1-digit operand, multiplication by a 2-digit operand (0.25), or a shift of 2 bit positions.

Thus, it can be said that a system evaluation process MUST have certain errors. If the description emphasizes functional data details, the errors may arise from misleading preliminary analysis of the application task; and if the description omits detail, the errors arise from the problem being incompletely defined (and therefore, analyst-dependent). The object, then, is to minimize these inherent errors without unduly complicating the estimating process.

Granted that these conditions hold true, it remains to be shown that the differences in levels of available detail do seriously affect the final timing estimate. This is not always obvious, nor have previous estimating procedures been concerned about variations in data that cannot be accounted for by use of medium or made values.

In considering the details analysis, it has been seen that any analysis will contain the following criteria in one form or another:

(1) The number of multiplications and additions.

(2) The file lengths, expressed in numeric and alphameric characters.

At looking at the arithmetic operations, it appears to be a fact that many computers will continue to be marketed without multiply and divide instructions as part of the standard instruction repertoire. This was originally the case because of the actual expense (and difficulty) of producing the necessary logic. Now, however, it appears to be based on reasons related to marketing economics. The same system with and without multiply and divide can be sold at two different prices, each to be evaluated by the user from an application viewpoint determined by the degree to which these or any other features are used by the application processes. In computer systems where hardware facilities have been omitted, subroutines or software packages are often supplied to carry out the functions. However, a competent systems analyst will always look at a multiplication to see if it can be replaced, not by the routine, but by simplified repetitive addition. Typically this may reduce the execution time by up to 80%. Thus, in a case which is central processor limited, and where half the processor load happens to come from multiply instructions, a firm specification of programming procedure may well lead to a timing estimate of 1 hour, where in fact, a systems analyst could produce an interpreted timing estimate of 36 minutes. This would result in an apparent error of 40%.

In another case, by expressing the file length in specific alphabetic and decimal characters, many systems decisions are implied. For instance, suppose that part of the file contains an address reference. Then two sets of decisions are implicit in the description:

(1) The amount of coding for the address (do the digits 191 stand for Philadelphia, Pa. ?); and

(2) Whether or not variable length fields are being utilized (e.g., does "Smith" take up 20 places in the name field because "McBornquesser - Smythe" may have to be accommodated in the field ?).

TABLE 1. VALIDITY OF EDITING ELEMENTS

Specification	Purpose	Rationale
EV 1205 - Number of index registers.	Used to establish the efficient methodology of addressing operands, from blocked records.	When I/O is limiting and is being minimized, a shortage of index registers may force constant unloading and reloading of the available registers. This may take 4 - 6 instructions, and require some 20 times the cost of indexing. The specific case is where I/O is dominant over a central processing loading primarily due to simple update operations. By minimizing the I/O and not recognizing the index costs, it is possible to be in error by some 50% of 100%, and to give a 25% incorrect dominant situation.
EV 1206 - Number of words that can be reached without indexing.	Used to establish the methodology of addressing operands when indirect and index possibilities exist, or to establish the ability of the index register to handle normal situations.	The size of error, and reason is as seen in EV 1205. The cause here may not be the shortage of index registers, but their comparative uselessness.
EV 1304 - Time to index an operand.	Used to establish the cost of addressing operands under storage-limited and similar conditions.	Where I/O is critical, but central processing is major, and consists primarily of simple update operations; this difference may add 50% to the cost of a simple update operation, therefore, its absence can give an error of 25%.
EV 1305 - Time to indirectly address an operand.	Used to establish the cost of addressing operands under storage-limited and similar situations.	This may be able to replace the index registers, and thus the error possibilities shown above. The error possibility is, therefore, the same as for EV 1205 and EV 1304 (i.e., 25%).
EV 1307 - The time involved in selecting a single digit from the center of a word in storage, and preparing it for use as a right-hand justified operand.	Used to prepare the cost of modification needed in obtaining operands under specific limiting conditions.	Forms 2 out of 11 items in the modifications due to I/O limited. When this modification is applied to the simple update task, it is applied 11 times within 5 tasks. Taking modification time as being as great as a task time, gives a maximum error of 25%. This, however, would not operate randomly, but in favor of specific systems, and therefore, while the time could be approximated, the overall ranking might be skewed.
EV 1309 - The time involved in moving an instruction from one portion of the store to another.	Used in the preparation of control per record timing.	This is not important in the case of very short, blocked records on the longest file. If each record must be handled separately, and if negligible computing occurs (a memory switching system perhaps) this could involve 1/3 of the central processor time. Where the central processor is operating on a character basis but the I/O is using a parallel buffer format, this percentage could be applied to the full system, thus giving a potential average error of 16%.
EV 1310 - The time involved in moving a record of N characters from one portion of the store to another.	Where appropriate, it is used to evaluate the cost of the input.	In the case where central processor and I/O usages are nearly equal, ignoring this element could provide errors in underestimating central processor time. The problem is related to the maximum number of I/O channels simultaneously in use, and the percentage of usage. One sample machine utilizes 13.3% of the transfer time in this operation. If, therefore, 4 channels were simultaneously in use, and by ignoring this factor, central processor and I/O were set equal, a maximum error of 53% would occur, or, using averaging methods, an error of 26% is unavailable in some circumstances.

In our first example, the resultant error was easy to evaluate. It came to 40%. In this case, it is not so easy to evaluate, but a reduction of 15 alphameric characters to 3 numeric digits, or of 20 to 5 characters, can hardly be ignored; particularly because these reductions may be cumulative, not compensating.

With these thoughts in mind, the accuracy limits of the VECTOR process were set at $\pm 10\%$. That is to say, any factor which might give rise to a difference of more than 10% in a performance timing was included, while other factors were ignored. It was intuitively considered that by this method the results could be held within an error range of $\pm 20\%$.

In computing these errors, one point of constant misunderstanding was noted whenever the project was being explained to outsiders. Our argument ran as follows: "Operations of type A must be distinguished from operations of type B because an error of greater than 10% could arise if the two were confused."

"Ah, but in the general case," replied the outsider, "operations of both types A and B account for only a small portion of the total processing time. Therefore, you can afford to ignore the distinction between them (or even to ignore the two types of operations completely)."

The point that was missed was that if there existed any reasonable set of specific circumstances, for any specific application, under which the lack of a distinction between the two types of operations could give rise to an error of greater than 10%, then the distinction would have to be made in order to meet the overall accuracy requirement for the VECTOR process. The "general case," in this context, is irrelevant.

All the elements in the Engineering Vector have been tested for validity. Most of the elements, such as the magnetic tape transfer rate, are obviously valid; but for others, the validity is less obvious. Table 1 shows the purpose and rationale for the elements that seem most likely to be queried.

2.4. VALIDITY OF THE EDITING ELEMENTS (EV 1401 - EV 1420)

It is not easy to understand intuitively why 20 distinct elements should be required in the Engineering Vector to describe a computer system's performance on input and output editing tasks. The following discussion will explain why these distinctions are essential.

Typically, a card or line image can be read from or written on magnetic tape in approximately 6 milliseconds if the images are unblocked, or 1 millisecond for efficiency blocked images. Also, typically, a line or card contains some ten fields. Combining these factors, we calculate input or output times of 600 or 100 microseconds per field in these circumstances, giving error thresholds of 120 or 20 microseconds, respectively.

Insofar as editing times varying from zero to 5,000 microseconds per field are being recorded in the medium priced commercial computer field (where it would be normal for these times to be minimized wherever possible), it is clear that some subdivision has to be made between varying types of editing so as to hold the error within reasonable limits.

For a distinction to be considered valid, it is necessary to show that it can cause a variation of at least 2 x 20 microseconds per field, and preferably 2 x 120 microseconds. (Doubling the error size is allowable here; if the average is taken between such limits, the error involved is half of the difference). The categories which have been chosen are:

(1)	Input	vs.	Output
(2)	General routines	vs.	Straight-line coding
(3)	Alphabetic data	vs.	Numeric data
(4)	Scientific numeric	vs.	Commercial numeric
(5)	Aligned to computer word structure	vs.	Not aligned to computer word structure

These will be considered separately under the quoted criteria:

2.4.1 Input Versus Output

The tasks an input edit may need which an output edit may not need include:

(1) Verification of validity of input data.

(2) Translation of input data to central processor code. +

(3) Testing to verify correct operation of the instruction.

+ This does not necessarily include radix conversion, but may involve merely changing the number of bits per character in the internal representation.

In some computers, this may involve separate character-by-character scans over the data for all of these purposes.

Considering our limit of 240 microseconds per field, this would put the cost of a single character scan at no more than 240/ (3 x 8*) = 10 microseconds. Such an achievement is not possible in non-character oriented systems —specifically, the Honeywell 800, with its 3-address instruction format and 6-microsecond memory cycle, does not approach such a figure. Thus, this distinction appears valid.

2.4.2 General Routines Versus Straight-Line Coding

While the result of the two processes may be identical, a very different logical approach is used when writing generalized routines as against specialized straight-line coding. A generalized editing routine must check for all the types of editing needed; it needs to be entered, to be supplied with the appropriate parameters and data, and to return the desired result to the main program. In addition, particularly in word-oriented systems, it often processes one character at a time, tests for end of routine, and (if not finished) obtains the next character and repeats the process. Effectually none of these steps need be taken by a straight-line coding of the same task. Therefore, for each n-character field being edited with m possible types of editing, the following extra operations may be needed:

> (Routine entry + routine exit + 2 routine parameter provisions + 2 data storages) plus m(tests of edit type needed) plus (n - 1)(2 move operations +1 count + 1 jump).

For a 5-character field being passed through a 4-option routine, this results in:

$$6 + m + 4n - 4 = 6 + 4 + 20 - 4 = 26 \text{ extra operations.}$$

Using the figure of 240 microseconds as the maximum permissible error, this allows less than 10 microseconds per operation. Many systems could not approach this; specifically, the Honeywell 800 with its 3-address instructions and 6-microsecond core cycle time.

---

* For this type of testing, the full input-output area is often used, rather than the selection of specific fields. A full length, therefore, is considered as 80/10 rather than 5.

2.4.3 Alphabetic Data Versus Numeric Data

Editing of alphabetic data does not need to involve translation into an internal code; editing of numeric data often does. Typically, a decimal to binary conversion involves a multiplication, an addition, a load register, and a test and jump for each character.

This would require some 20 simple instructions plus 5 multiplies per 5-character field. Assuming a 3:1 ratio between the multiply and addition times, an addition time of 7 microseconds and a multiply time of 21 microseconds would be required within the allowable error. Specifically, none of the three sample systems are able to approach these figures.

2.4.4 Scientific Numeric Editing to Commercial Editing

Editing of scientific numeric data does not normally call for more than sign insertion and zero suppression. Both of these are often hardware capabilities, allowing the whole field to be handled as a unit.

By contrast, none of the sample machines are able to handle automatically any of the other COBOL editing requirements: CR or DB insertion, comma and decimal point insertion, floating dollar or check protect operations. Typically, these are done on a character-by-character basis; each character is retrieved and stored separately and tested as to its position and required type of edit (out of perhaps five possible types). Furthermore, some housekeeping is involved for each character.

This gives a figure of 10 operations per character, or 50 operations per 5-character field — none of which are required for scientific editing. Using the 240 microsecond limit, all examined systems would have to be able to handle these operations at a rate of 4.8 microseconds per operation or faster. Specifically, none of the three sample systems could match this rate. Therefore, this distinction is valid.

2.4.5 Aligned with the Computer Structure Versus Not Aligned with Computer Structure

To pick up a field and prepare it for editing, or to place it into a specified location subsequent to editing, requires a load register instruction, a shift of n characters, a load register instruction, and a shift of m characters. Some computer systems only have shifts

in one direction, so that the total shift involved may be up to 2 computer words long. Typically, this takes a memory cycle per shift, so that for a 48-bit system it would be possible to use 94 memory cycles in this process. In such a case, any memory cycle time of 3 microseconds or more (and there are many of these) could exceed the 240 microsecond limit. In fact, none of the three sample systems will need this length of time, but the H-800 does require 196 microseconds for the operation. This is greater than the lower threshold (48 microseconds) of error factor, but could possibly be ignored in gross estimating.

2.4.6 Summary

Of the five suggested criteria for establishing different edit times, all were able to produce differences which would cause errors of greater than 20% if averaged figures were used when the file concerned was dominant and blocked.

All except one (aligned or not aligned with the computer word structure) were able to produce differences which would result in errors of greater than 20% if averaged figures were used when the file concerned was dominant and unblocked. The aligned/not aligned differentiation was in these circumstances able to produce an error of 17%.

2.5 TESTING THE VECTOR PROCESS

The graphs (Figures 20, 21, 22) show that the VECTOR process is producing lower times than those published in AUERBACH Standard EDP Reports. Moreover, the VECTOR times are consistently lower. This fact requires some examination.

AUERBACH Standard EDP Reports is, at the moment, as far as we know, the only source of timing figures for the performance of a large number of computer systems on certain, specified problems. In the Users' Guide, the problems are described in detail, flow-charts are laid out, and standards are set, so that the analyst has few (if any) policy decisions to make. Blocking factors are preset and, in many ways, programming policy is laid down ahead of time.

The analyst then codes and times, separately, each portion of the program — the time per transaction record, the time per tape block, etc. — without considering the ratio between the transaction and master files. Naturally, he uses such programming methods as seem advisable to him under these circumstances.

GENERALIZED FILE PROCESSING

Test Problem A, estimated by VECTOR Process and by AUERBACH Standard EDP Reports

Data Record Sizes
- Master File: . . . . 108 characters.
- Detail File: . . . . 1 card image.
- Report File: . . . . 1 line image.

Computation: . . . . standard.

Time Basis: . . . . using estimating procedure outlined in Users' Guide in AUERBACH Standard EDP Reports.
as estimated by VECTOR Process, (see Worksheet in Technical Note 5).

Graph: . . . . see graph below.

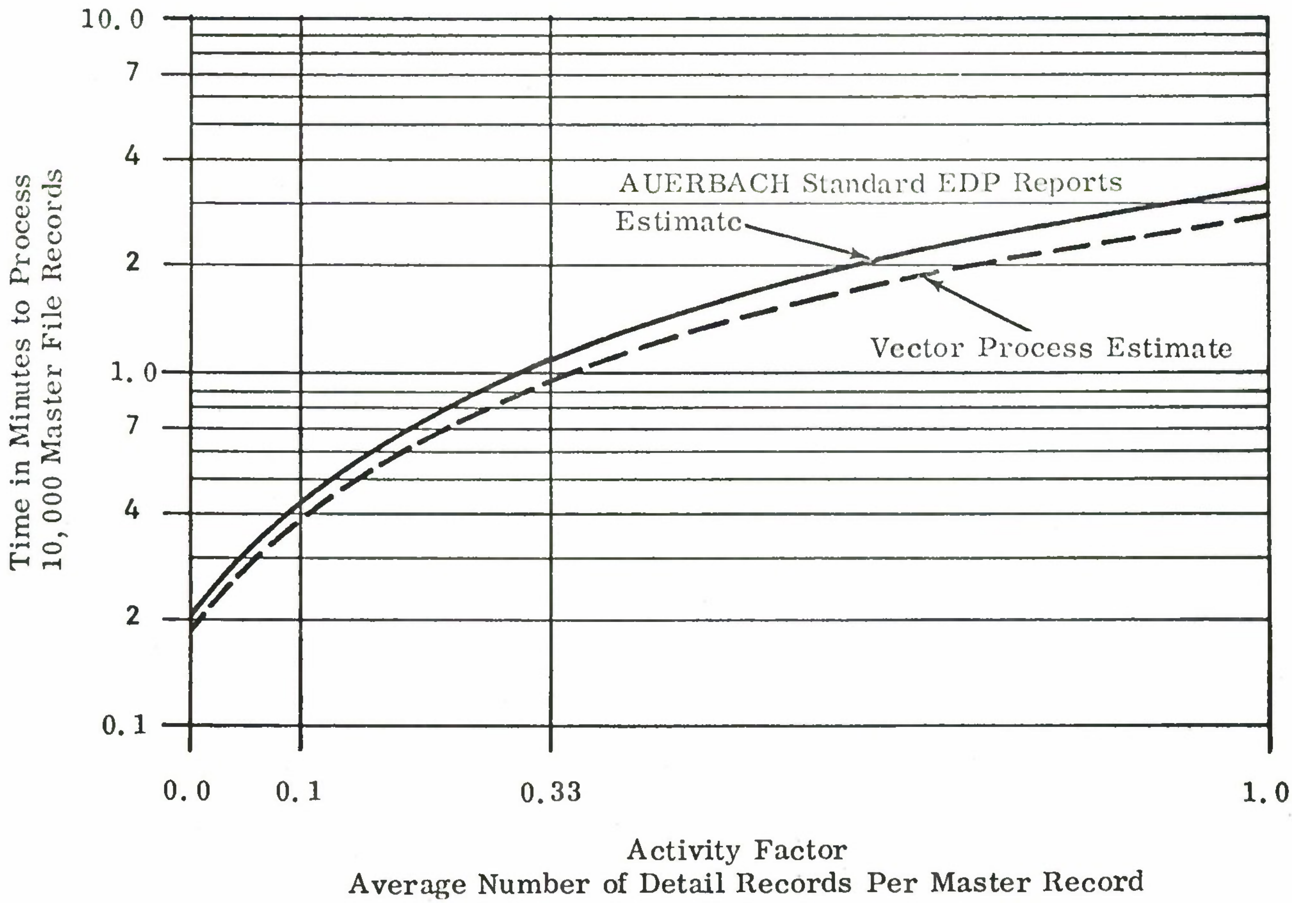


Figure 20

COMPARISON OF SYSTEM PERFORMANCE ESTIMATES FOR IBM 7074 COMPUTER SYSTEM

GENERALIZED FILE PROCESSING

Test Problem A, estimated by VECTOR Process and by AUERBACH Standard EDP Reports

Data Record Sizes
- Master File: . . . . 108 characters.
- Detail File: . . . . 1 card image.
- Report File: . . . . 1 line image.

Computation: . . . . standard.

Time Basis: . . . . using estimating procedure outlined in Users' Guide in AUERBACH Standard EDP Reports.
as estimated by VECTOR Process, (see Worksheet in Technical Note 5).

Graph: . . . . see graph below.

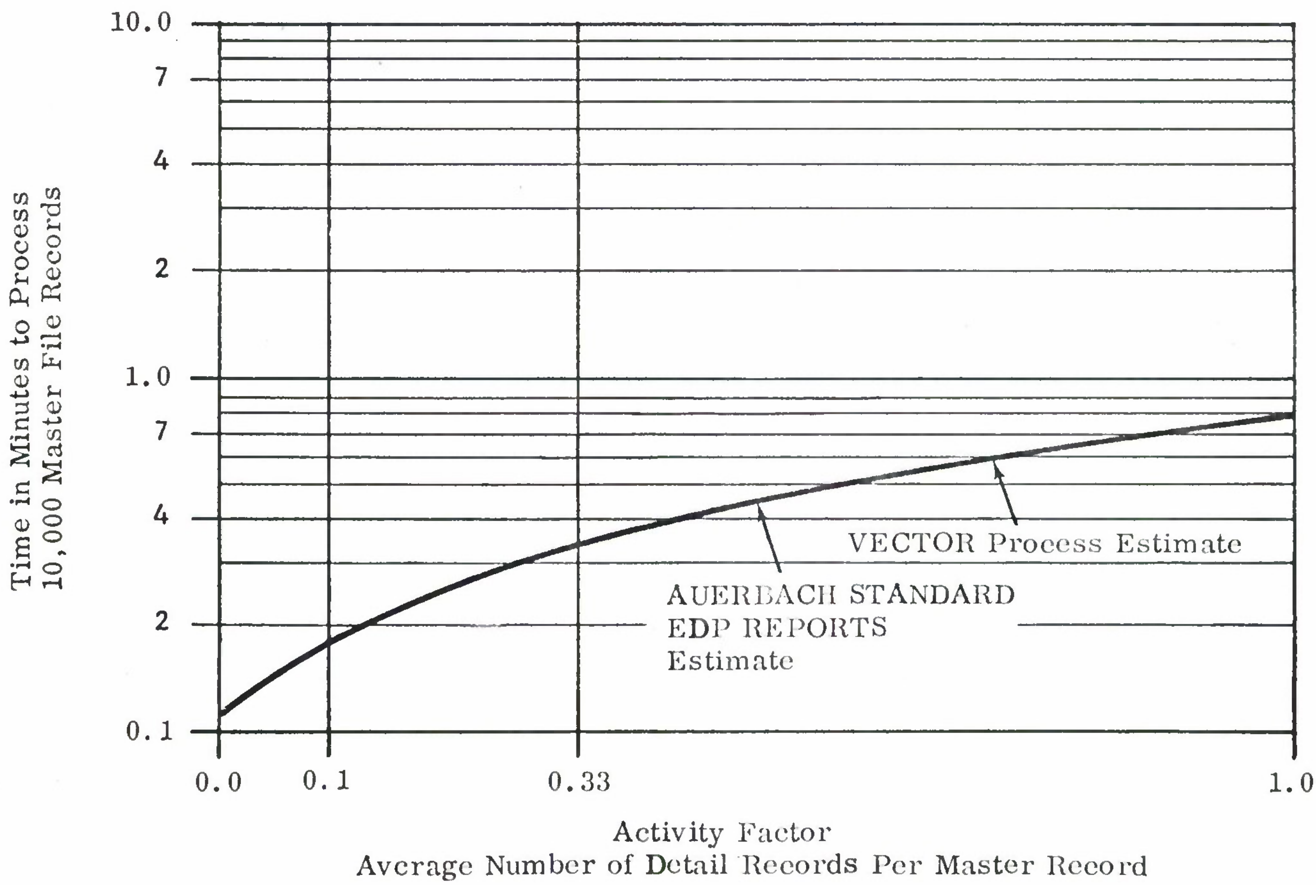


Figure 21.

COMPARISON OF SYSTEM PERFORMANCE ESTIMATES
FOR UNIVAC III COMPUTER SYSTEM

GENERALIZED FILE PROCESSING

Test Problem A, estimated by VECTOR Process and by AUERBACH Standard EDP Reports

Data Record Sizes
- Master File: . . . . 108 characters.
- Detail File: . . . . 1 card image.
- Report File: . . . . 1 line image.

Computation: . . . . standard.

Time Basis: . . . . using estimating procedure outlines in Users' Guide in AUERBACH Standard EDP Reports.
as estimated by VECTOR Process, (see Worksheet in Technical Note 5).

Graph: . . . . see graph below.

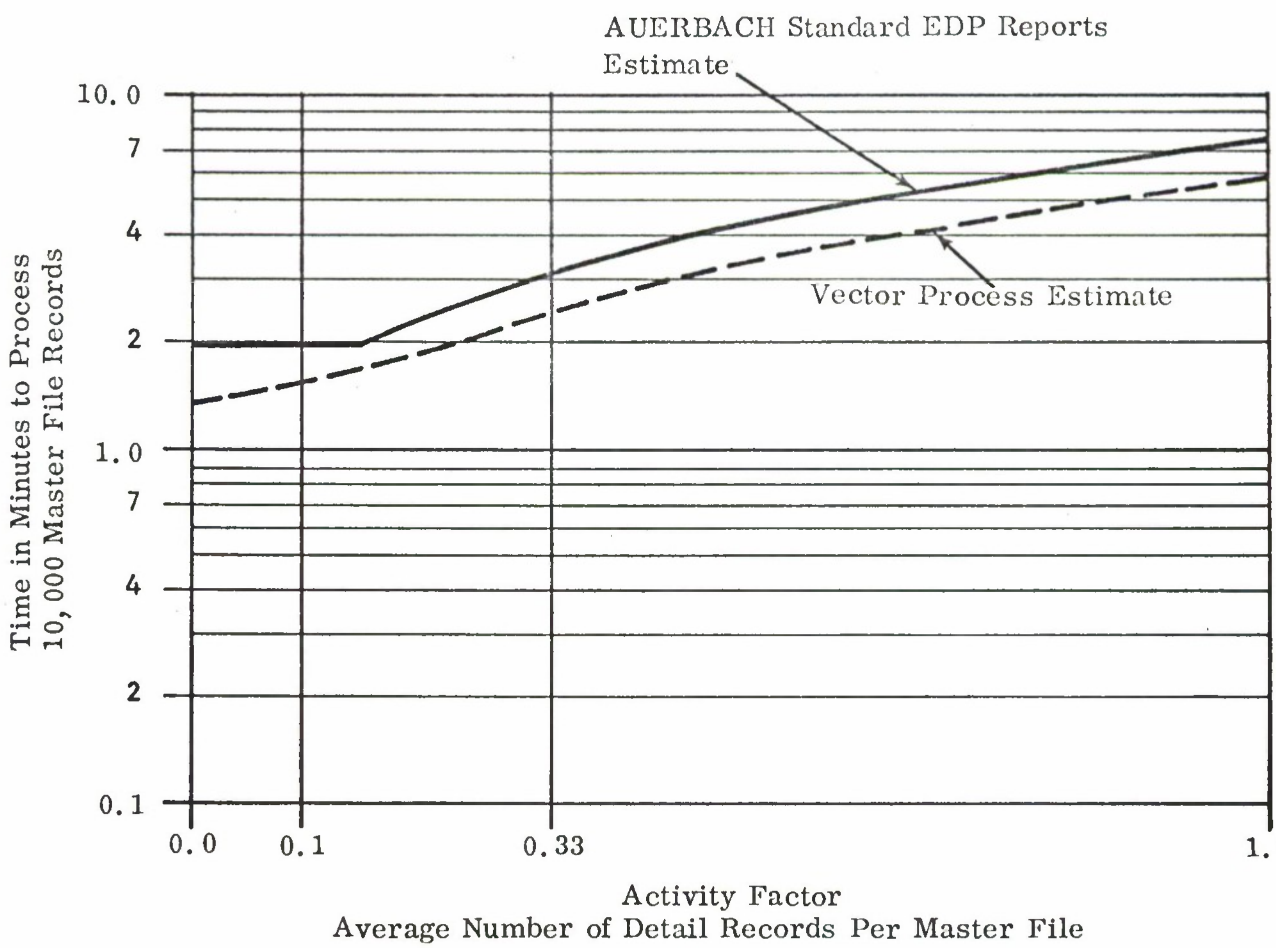


Figure 22.

With these figures, he then evaluates the timing for a number of different activity ratios, and draws the published graph and prepares worksheets. It is assumed that anyone who wishes to use the published data to estimate the time required for his own problem on a specific computer will start with the worksheet figures, consider their implications, *amend them* in accordance with the data in the rest of the report, and then, following the standard procedure laid down in the Users' Guide, work out his own estimates.

It is *not* anticipated that he will just read and extrapolate from a single graph to his own problem. In short, the AUERBACH estimates are strictly *comparable* estimates arrived at in a specified way which is suitable for easy amendment to other problems if more detail is required.

The aim of the VECTOR process is very different. A VECTOR estimate is only concerned with a single point, not a graph of a whole range. It, therefore, adjusts the blocking factors, programming procedures, etc., to fit that specific point without concern as to what the effects might be for other activities. This is done deliberately in order to provide the user with what he is interested in: the performance he can expect on that specific application.

These adjustments naturally reduce the estimated time, as they are intended to; and to some extent it may be thought that the difference between the two figures is an estimate of the potential gross profit which can be obtained through efficient systems design on the specific application mix. This is purely an intuitive reaction, however, and has not been investigated.

Within the basic premise that the VECTOR process should produce estimated times below those of *AUERBACH Standard EDP Reports*, the results appear to be quite consistent over the whole range of the test problem. Timings were calculated for 11 separate points, compared with *AUERBACH Standard EDP Reports*, and the variations noted. The results are listed in Tables 2, 3 and 4. The consistency of the figures gives preliminary support to the thesis that the VECTOR process timings are valid.

TABLE 2

COMPARISON OF ESTIMATES PROVIDED BY VECTOR PROCESS WITH THOSE PUBLISHED IN AUERBACH STANDARD EDP REPORTS

PROBLEM: AUERBACH Standard File Problem A on Honeywell 800.

VECTOR ESTIMATE: Working Papers in TN5.

AUERBACH STANDARD EDP REPORTS ESTIMATES: Extracted from Report Dated April, 1963 on Honeywell 800, Configuration VIIIB.

ACTIVITY	VECTOR ESTIMATES		VECTOR ESTIMATES		% DEVIATION	APPARENT REASON FOR DEVIATION
	Time, Minutes	Limiting Factor	Time, Minutes	Limiting Factor		
0.0	0.18	I/O	0.20	I/O	-10	Vector uses larger blocking factor.
0.1	0.38	CP	0.42	CP	- 9.5	Use of specially prepared straight line editing routines for input and output.
0.2	0.59	CP	0.72	CP	-18.0	
0.3	0.85	CP	1.0	CP	-15.0	
0.4	1.11	CP	1.3	CP	-15.0	
0.5	1.4	CP	1.6	CP	-12.5	
0.6	1.6	CP	1.8	CP	-11.1	
0.7	1.9	CP	2.2	CP	-13.6	
0.8	2.15	CP	2.5	CP	-14.0	
0.9	2.4	CP	2.8	CP	-14.3	
1.0	2.7	CP	3.1	CP	-12.9	

CP = Central Processor limited.
I/O = Input/Output limited.

TABLE 3

COMPARISON OF ESTIMATES PROVIDED BY VECTOR PROCESS WITH THOSE PUBLISHED IN AUERBACH STANDARD EDP REPORTS

PROBLEM: AUERBACH Standard File Problem A on IBM 7074

VECTOR ESTIMATE: Working Papers in TN5.

AUERBACH STANDARD EDP REPORTS ESTIMATES: Extracted from Report Dated November 1962 on Configuration VIIIB

ACTIVITY	VECTOR ESTIMATES		ASEDPR ESTIMATES		% DEVIATION		APPARENT REASON FOR DEVIATION
	Time, Minutes	Limiting Factor	Time, Minutes	Limiting Factor			
0.0	0.11	I/O	0.17 (0.11)	I/O	-35	(0)	VECTOR blocked Master Files more efficiently than ASEDPR.
0.1	0.17	CP	0.17 (0.17)	I/O	0	(0)	
0.2	0.24	CP	0.34 (0.23)	I/O	29	(+4)	VECTOR blocked report and translation files more efficiently than ASEDPR. No significant deviation could be found between ASEDPR and VECTOR central processor times.
0.3	0.30	CP	0.51 (0.30)	I/O	41	(0)	
0.4	0.37	CP	0.67 (0.36)	I/O	45	(+3)	
0.5	0.43	CP	0.84 (0.42)	I/O	49	(+2)	
0.6	0.50	CP	1.01 (0.49)	I/O	50	(+2)	
0.7	0.56	CP	1.17 (0.55)	I/O	52	(+2)	
0.8	0.63	CP	1.35 (0.61)	I/O	53	(+3)	
0.9	0.69	CP	1.51 (0.68)	I/O	54	(+1)	
1.0	0.75	CP	1.68 (0.73)	I/O	55	(+3)	

CP = Central Processor limited.
I/O = Input/Output limited.
ASEDPR reflects the timing of the output the Report File on tape done without blocking.
Figures in brackets show central processor times which would be attainable given sufficient blocking on the files.

TABLE 4

COMPARISON OF ESTIMATES PROVIDED BY VECTOR PROCESS WITH THOSE PUBLISHED IN AUERBACH STANDARD EDP REPORTS

PROBLEM: AUERBACH Standard File Problem A on Univac III.

VECTOR ESTIMATE: Working Papers in TN5.

AUERBACH STANDARD EDP REPORTS ESTIMATES: Extracted from Report dated March 1963, on Univac III, Configuration VIIIB.

ACTIVITY	VECTOR ESTIMATES		ASEDPR ESTIMATES		% DEVIATION	APPARENT REASON FOR DEVIATION
	Time, Minutes	Limiting Factor	Time, Minutes	Limiting Factor		
0.0	0.12	I/O	0.20	I/O	-40	Increased blocking on main file.
0.1	0.15	I/O Note 1	0.20	I/O	-25	Special attention is given to minimizing handling of non-active records. Specifically, index registers are not used.
0.2	0.17	CP	0.22	CP	-23	
0.3	0.22	CP	0.28	CP	-21	Arithmetic operations are handled with minimum house-keeping. Data records are held so as to be more easily handled in the system.
0.4	0.27	CP	0.35	CP	-23	
0.5	0.31	CP	0.40	CP	-22	
0.6	0.36	CP	0.48	CP	-25	
0.7	0.41	CP	0.53	CP	-23	
0.8	0.46	CP	0.60	CP	-23	
0.9	0.51	CP	0.68	CP	-25	
1.0	0.56	CP	0.73	CP	-23	

CP = Central Processor limited
I/O = Input/Output limited.

Note 1 - At activity 0.1, the smallest time is given by minimizing Central Processor time. However, at this point where CP is effectively minimized, the limiting factor is the I/O.

SECTION 3. FURTHER STUDIES

3.1 AREAS OF FURTHER STUDY

The present research has led to a comprehensive system for estimating the running time for a specified data processing run on a given computer. The accuracy of this method appears to be adequate for evaluation purposes. However, certain restrictions in the scope of the method had to be enforced by the limitations of time and personnel on the project thus far. The work has also suggested other areas of study pertinent to computer performance evaluation. There are, therefore, several directions for further studies such as the following:

(1) Remove the current restrictions so that the evaluation procedures are broader in scope. In particular, the procedures can be expanded to cover more types of applications and more types of computers, to estimate storage requirements and to include the effects of environmental factors.

(2) Perform further tests. The present and expanded procedures should be tested for validity. Investigations of the sensitivity of the results to various parameters, should be undertaken.

(3) Incorporate methods of including software development costs and software performance in the VECTOR procedure.

Possible specific further studies are described in the remainder of this section.

3.2 EXPANDED APPLICATION SCOPE

At present we have considered exclusively tape file-maintenance runs. Studies should be undertaken to expand the area of application to include:

(1) Problems requiring random access and, therefore, implying storage of files on mass-access (e.g., disc) units.

(2) Programming aid runs, such as assembly, testing, and compilation runs.

(3) Scientific and engineering applications.

Extension of the present methods to random access systems and to the programming aid runs appears to be straightforward.

The problem of evaluating computers for scientific and engineering applications is important. It is not clear, however, whether the present techniques are directly applicable to these areas. Further studies, starting from the present viewpoint, might lead to fruitful results. Sorting can be classed as a scientific procedure and work on it is not of great importance since methods of estimating sorting time for existing sorting methods are generally available and straightforward.

3.3 EXTENSION OF EQUIPMENT SCOPE

The Engineering Vector and the evaluation procedures should be expanded to include:

(1) Peripheral equipment other than magnetic tape, such as card readers, check sorter inputs, data-links, inquiry units, etc.

(2) Main-satellite workload allocation. The present method does not specifically take into account the possibility of varying the distribution of tasks (proceeding simultaneously) between a main and a satellite computer (on-line or off-line).

The extension to other peripheral devices appears to be a straightforward extrapolation of the present method.

The problem of proper distribution of a job between a main and satellite computer could be solved by repetitive use of the VECTOR process, but further study might result in a more straightforward allocation procedure.

3.4 CONFIGURATION VARIATIONS

In preparing the VECTOR process, a number of approaches to the automatic optimization of configuration requirements were examined. While some of these were found to be valid, the volume of computation was high. Furthermore, the value of such an automatic computation may be low since a human (with a small amount of background information) examining the results of a selected configuration, can intuitively go directly to the next configuration to be tested. That is, he can generally choose the configuration increment which increases the performance/cost ratio the most.

Therefore, one direction of development is to provide the human analyst with sufficient background data about the performance on a given configuration to make the proper configuration decisions; for example, data about the sensitivity of performance to changes in tape speed, storage capacity, etc.

Further research studies could conceivably develop specific configurations optimization procedures which were relatively simple. The question of configuration selection, however, is combinational and it appears difficult to avoid situations in which many of the possible configuration combinations have to be tried in order to select the best one.

3.5 STORAGE REQUIREMENTS

The Vector specifications at present make it possible to estimate apparent storage requirements. This estimating procedure, however, is not thought to be sufficiently accurate, nor have realistic tests been undertaken to estimate the accuracy.

One of the problems in estimating storage requirements is that it appears to be sensitive to the degree of knowledge possessed by the person preparing the Functional Specifications. The procedures which lead to those Functional Specifications which are used to compute storage requirements should be made more specific. In other words, to obtain accurate storage estimates the user may have to have more data about the specific run than is normally available at the time of computer evaluations.

3.6 ENVIRONMENTAL CONSIDERATIONS

There are environmental factors surrounding a computer installation which must be considered in an overall evaluation. Sufficient detail is presently built into the VECTOR process to allow rough evaluation of these effects; also the evaluation of environmental considerations may depend on a knowledge of the analyst about the people who will operate the system, and this input requires further definition.

3.7 FURTHER TESTING

Tests run to date on the VECTOR process have been designed to check the validity of the system itself rather than to estimate the accuracy of the output. Timing produced by VECTOR has been compared with timings produced by analysts with similar

backgrounds but working independently and in detail. Two further tests should be undertaken before the method is considered for practical applications.

(1) Validity Tests - The system should be validated by use in actual situations(e.g., RFP's) which have been timed out by other means. For example, a review of past equipment manufacturers' proposals for a specific data processing run would provide data for comparative timings.

A test might be made by attempting to forecast proposal timings in selected cases. Of course, the ground rules for such a test must be carefully worked out. This test would be particularly valuable if it were possible to request proposals from any supplier who looked promising under the VECTOR analysis, whether or not there was originally an intention to invite proposals from these suppliers. A final test might be obtained by evaluating actual performance in the field. The actual running times could be compared to those predicted by VECTOR from a description of the run.

(2) Sensitivity Analysis - During the development of the model, a number of assumptions were made. The guiding principle in making these assumptions was that errors of greater than $\pm 20\%$ in the final process were not desirable. For instance, it was considered that errors resulting from assuming that all numeric operands were 5 digits long, would be less than 20%. However, it was considered that assuming all magnetic tape file blocks were a thousand characters would involve significant errors, and such an assumption was not made.

These many assumptions were made without reference to any particular data but based on general background of the researchers. If possible to obtain data (for example, from detailed proposals or actual installations), these assumptions should be re-evaluated. In general, the sensitivity of the output of VECTOR to certain key parameters and specifications, is an area for further study.

3.8 METHODS OF INCLUDING SOFTWARE COST AND SOFTWARE PERFORMANCE INTO VECTOR

It is clear that the cost of programming and of operating a processing run is greatly affected by the software available; software to implement the operational program, software to assist in the programming process, and software involved in the operating environment. The present method assumes that the evaluator will adjust the estimates to take into account software in an appropriate manner. We believe, however, that further study would lead to the development of a specific procedure which could give proper weight to the existence of various programming packages. Unfortunately, the effectiveness of software packages depends to a large extent on the way they are implemented in any particular installation, so that estimating performance which includes software, is difficult at best.

The collection and verification of the data for such a system will involve a mixture of study of actual case histories and the estimates on the part of programming experts.

A most critical software problem is to estimate the effect of source language-compiler systems on programming time and, in particular, on operating time. It is believed that a method can be developed which would permit such estimates to be made. The general approach would be to define a number of "macro procedures" of the type normally found as verbs in common source languages. The time required to run a macro after having been compiled in various languages, would then be investigated. This should lead to a method of developing overall timing estimates for actual applications specified in terms of the macros involved.

AUERBACH Corporation has investigated the estimation of software costs and performance on another project, and has had some success in developing standardized estimating procedures.

APPENDICES

APPENDIX I

GUIDE AND FORMS FOR COMPLETION OF TRANSFORMED ENGINEERING VECTOR OF THE "VECTOR" ESTIMATING PROCESS

APPENDIX I

GUIDE AND FORMS FOR COMPLETION OF TRANSFORMED ENGINEERING VECTOR OF THE "VECTOR" ESTIMATING PROCESS

1. INTRODUCTION

The VECTOR method uses independent sets of numbers (Vectors) to represent the important characteristics of each computer and each application. The Vectors are used both to establish and time the optimum method on a specific computer system. The Vectors themselves are normally self-sufficient and no other knowledge of either the problem or the computer is assumed.

In these circumstances it can be seen that the creation of these Vectors is an unusually responsible process. This document is concerned with the preparation of the Computer Vector, shows the procedural steps needed, and attempts to illustrate the requirements and the opportunities given to allow each computer system to be shown at its honest best.

2. COMPUTER DEFINITION PROCEDURE

The first step is to define the basic characteristic of the computer by filling in the best possible answers to the applicable queries on the Engineering Specification form, which is shown and fully described in Section 3. (It is anticipated that this form will normally be filled in by a representative of the computer manufacturer who has access to all the relevant facts and knowledge of the most efficient techniques for programming and operating the equipment.) The estimated performance of each computer system will be directly related to the answers supplied on the Engineering Specification form, so it is of utmost importance that each applicable query be answered as completely and realistically as possible.

The second step in the computer definition procedure is the conversion of the raw Engineering Specification into a standardized form known as the Engineering Vector by application of the Engineering Conversion Algorithm. This step, and all subsequent ones, will normally be performed by the people using the VECTOR method. Whereas many items of the Engineering Specification are applicable only to certain basic types of computer systems (e.g., binary fixed word-length or alphanumeric character-oriented), each element of the Engineering Vector has the same meaning for every digital computer system. The Engineering Conversion Algorithm essentially consists of:

(1) Checking each element of the Engineering Specification for reasonability, proper format, and correct units, and making any necessary corrections.

(2) Direct transfer of items from the Engineering Specification to become elements of the Engineering Vector without alteration.

(3) Performing the specified simple mathematical operations upon the type-dependent items of the Engineering Specification to produce corresponding standardized elements in the Engineering Vector.

The exact operations required to produce each element of the vector are specified on the Engineering Vector form, which is shown and fully described in Section 5. The Engineering Vector for each computer system is a set of 60 numeric quantities, each of which has a standardized, precisely defined format, derivation, and significance. It is believed that these 60 elements provide all the information about the computer system that is needed to produce objective timing estimates for file processing applications on magnetic tape-oriented systems.

Although the Engineering Vector summarizes all the relevant characteristics of a particular computer system, it is not yet in the most efficient form for convenient combination with the related functional vector elements. This combination of numbers is achieved in the Performance Algorithm. To minimize the amount of work that must be done to produce each individual timing estimate, the Engineering Vector is transformed into a more suitable form by pre-computing certain numerical quantities that would

otherwise have to be computed each time the Performance Algorithm is executed. This third procedural step is called the Engineering Transformation Algorithm. Its output is the Transformed Engineering Vector, which can be stored in a library until it is needed for a specific performance estimate.

The Transformed Engineering Vector actually consists of four distinct sets of processed computer characteristics. Each set defines the maximum performance of the computer system under one of the following limiting conditions:

(1) General (no known limitation).

(2) Central Processor Limited, with sufficient storage space.

(3) Input-Output Limited, with sufficient storage space.

(4) Storage Space Limited.

The use of these "cases" is a recognition that an analyst, in fact, "trades" input-output time for space, or programming time for running time wherever such trades help to improve the performance of his computer. Throughout the specification process, running times will be sought for specific tasks under such conditions as with "minimized object time." Naturally it is possible that these will be the same, but normally, they will differ. The amount of this difference will indicate just how much trade off is advisable for the system concerned. If this factor is considered when the various quantities are being computed, little difficulty should be found in understanding exactly what value is needed.

3. ENGINEERING SPECIFICATION

The specification of the engineering characteristics is divided into six parts, each of which will be considered separately. Table 1 shows the relationship between the parts. The basis for the separation is to allow a revision of one part to be made without any resultant changes in another part.

Part 1 of the Engineering Specification has been reserved for a description of background details such as the operational status of the computer system and the availability of assemblers, compilers, executive routines, and utility routines. This information will not be utilized in the present model of the VECTOR procedure, but will merely be collected and presented to the user to assist him in placing each computer system within the proper frame of reference.

Part 2 of the Engineering Specification deals with the fundamental hardware description of the computer itself. The questions here, and throughout the rest of the specification, are arranged as shown in Figure 1.

SPECIFICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 201	x	x		x		Main memory size in words.	words	
ES 202		x		x		Word size in alphanumeric characters.	chars	

Figure 1. Format of Parts 2 through 6 of the Engineering Specification

The first column gives a reference ES XX, standing for Engineering Specification No. XX.

The next five columns are headed:

Bin. FW, indicating a computer which operates normally with binary operands.

Dec. FW, indicating a computer which operates normally with 4-bit coded decimal digits, and whose arithmetic unit handles one word at a time.

Dec. Char., indicating a computer which operates normally with 4-bit coded decimal digits, and whose arithmetic unit handles one digit at a time.

AN FW, indicating a computer which operates normally with 6-bit characters, each character position capable of storing any alphanumeric symbol, and with one word (of a number of character positions) at a time.

AN Char., indicating a computer which operates normally with 6-bit character, each character position capable of storing any alphanumeric symbol, and whose arithmetic unit handles one character at a time.

This division into computer types was found advisable because many of the questions might be misinterpreted if they were not phrased directly in terms of the appropriate computer type. This arrangement avoids these problems as, once the appropriate computer type has been selected, only those questions which are checked in the appropriate column are answered.

The other three columns are headed "QUERY," "ANSWER," and "COMPONENT NO." The last column permits the inclusion of a reference to any comment that the specifier wishes to make about the answer he has just provided. Perhaps he might wish to include documentation, explain a formula, or note some other circumstance which might affect an answer. In all these cases he is urged to make full use of this comment column as a guard against subsequent misinterpretations.

The actual questions shown in Part 2 are simple and straightforward (see example below), and no difficulty is anticipated except where a feature is optional. This should be indicated in the comment column.

ES 203		x				Word size in decimal digits.	digits	
ES 204		x				Word size in bits (excluding check bits).	bits	
ES 205					x	Main memory size in characters.	chars	
ES								

Figure 2. Example of queries in Part 2 of the Engineering Specification

Part 3 of the Speeifieation is also very straightforward. The queries (see example) ask for times whieh normally will not vary. The work to be timed is explained in detail on the Speeifieation form, and it is felt that no diffieulty should be found here.

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES [illegible]	x [illegible]					[illegible] is taken f[illegible] the fo[illegible] [illegible] its of the [illegible] ES 310 was zero, [illegible] ll be zero also.)		
ES 313	x	x	x	x	x	Time taken to inerement and test an index register.	μsec.	
ES 314	x	x	x	x	x	Time taken to move an instruetion from one part of the main memory to another loeation.	μsec.	

Figure 3 Examples of queries in Part 3 of the Engineering Speeifieation

Part 4 is less straightforward. This part deals with tasks whieh ean often be done in a number of ways. Consider the example shown below. It deals with a very neeessary task: reading a eard image and preparing it for the eomputer's use. This must be done eonstantly and may well be a major system eonsideration. But how should it be done? There are many ways, some of whieh will be used on one system, some on another.

* General input editing task: Take a field stored in main memory in punehed eard eode; verify the legality of the punehing; translate as needed; and unpaek so that the field ean be used direetly as an arithmetie operand.

Figure 4. Task from Part 4 of the Engineering Speeifieation

We have attempted to isolate the main items which will change the time (and method) a prudent systems analyst might use, and then ask each question for each possible case; i.e., every possible permutation of relevant circumstances under which the task might have to be performed has been included as a separate question.

This may appear to be an unnecessary detail; however, the fact remains that these circumstances, when critical, will change the performance of the system. If each computer is to be allowed to show its real strengths, every query is necessary.

ES 401	x	x		x		General input editing task* with programming minimized and 11-character alphabetic field. Input field is synchronized (i.e., aligned in accordance with computer word structure).	μsec.	
ES 402	x	x		x		General input editing task* with programming minimized and 5-digit numeric field. Input field is synchronized.	μsec.	
ES 403	x	x		x		General input editing task* with object time minimized and 11-character alphabetic field. Input field is synchronized.	μsec.	
ES 404	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is synchronized.	μsec.	

Figure 5 Four of the 12 queries in the Engineering Specification dealing with the General Editing Input Task

Part 5 is a straightforward series of questions regarding the computer system's possibilities for simultaneous operations. This has been designed for a tape-oriented system and does not attempt to cover the possibilities with other I/O units. It does cover the number and type of I/O channels, and the relationship (if any) between processing and I/O operations.

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 501	x	x	x	x	x	Number of magnetic tapes which can be reading with processing proceeding.		
ES 502	x	x	x	x	x	Number of magnetic tapes which can be reading with no processing proceeding.		

Figure 6. Examples of queries in Part 5 of the Engineering Specification

The last part of the specification deals with the magnetic tape units. Query No. ES 602 (see below) may be misunderstood, as it is phrased in an unusual way in order to produce a comparable answer whether or not the magnetic tape has to be slowed down or stopped between blocks. The query is simply "How many characters could be transferred, at peak speed, during the minimum amount of time it will take the tape unit to pass the tape from one end of a block gap to the other?"

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 601	x	x	x	x	x	Peak speed, in alphanumeric characters per second.	char/sec.	
ES 602	x	x	x	x	x	Cost in characters of a tape gap when passed over as quickly as possible.*		

Figure 7. Examples of queries in Part 5 of the Engineering Specification

4. ENGINEERING CONVERSION ALGORITHM

The primary function of this algorithm is to standardize the information in the Engineering Specification, so that directly comparable values will be produced for all types of computers. At the same time, a technical edit is performed to ensure that the specification is complete and accurate. Input for this purpose comes from the comments column and from the technical understanding of the editor.

The process is straightforward and self-explanatory. In most cases, the answer to a specified query in the Engineering Specification is examined to insure that it is complete, reasonable, and expressed in the proper format and units; then it is simply copied into the RESULT column of the Engineering Conversion Algorithm form. (For example, Figure 8 shows that the value of EV 1304 is identical with the values of ES 306 in the Engineering Specification.) In other cases, the evaluation of a formula with specified values for all parameters is required (e.g., EV 1301, in which the formula for addition time in a computer capable of using variable-length operands is evaluated for 5-digit operands).

5. THE ENGINEERING VECTOR

This is the central part of the computer description. The Engineering Vector is prepared from the Engineering Specification (and, in fact, the sole purpose of the Specification is to prepare it) and then, after transformation into a more convenient format, used in the Performance Algorithm. Thus, a computer is, for the purpose of this procedure, no more and no less than its Engineering Vector.

The present form of the Engineering Vector is illustrated in Figure 9. The first section of the Specification remains unchanged, giving the background of the computer system in readiness for review by a prospective user of the system. The subsequent sections reflect, section by section, the parts of the Engineering Specification. Thus, Part 2 describes the actual hardware, Part 3 describes the performance of the arithmetic unit on specific tasks such as addition, Part 4 shows the performance on input and output editing, Part 5 shows the capabilities for simultaneous operations, and Part 6 provides details on magnetic tape input and output.

The value of each element in the Engineering Vector will be a single numerical quantity.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT* µsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1301 Addition time	x	x	x	x	x	(ES 301), evaluating any formula with: No. of digits = 5. No. of significant digits = 5.		
EV 1302 Multiplication time	x					(ES 304), evaluating any formula with: No. of "1" bits in either operand = 8.		
		x	x	x	x	(ES 304), evaluating any formula with: No. of digits in either operand = 5. Sum of digits in either operand = 25. Complement of digits in either operand = 20.		
EV 1303 Division time	x					(ES 303), evaluating any formula with: No. of "1" bits in either operand = 8.		
		x	x	x	x	(ES 304), evaluating any formula with: No. of digits in either operand = 5. Sum of digits in either operand = 25. Complement of digits in either operand = 20.		
EV 1304 Indexing time	x	x	x	x	x	(ES 306)		
EV 1305 Time to increment and test an index register	x	x	x	x	x	(ES 313)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

Figure 8. Specimen Page from Engineering Conversion Algorithm.

ENGINEERING VECTOR

ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE
1201	Memory size in alphanumeric characters	
1202	Memory size in decimal digits	
1203	Word size in alphanumeric characters	
1204	Word size in decimal digits	
1205	Number of index registers	
1206	Volume of memory (in alphanumeric characters) that can be accessed without indexing	
1301	Addition time	
1302	Multiplication time	
1303	Division time	
1304	Indexing time	
1305	Time to increment and test an index register	
1306	Indirect addressing time	
1307	Unpacking time	
1308	Packing time	
1309	Instruction move time	
1310	Character move time	
1311	Single comparison time	
1401	Input edit: programming minimized, alphanumeric, synchronized	
1402	Input edit: programming minimized, numeric, synchronized	
1403	Input edit: object time minimized, alphanumeric, synchronized	
1404	Input edit: object time minimized, numeric, synchronized	
1405	Input edit: programming minimized, alphanumeric, not synchronized	

Figure 9. Specimen Page from the Engineering Vector

6. THE ENGINEERING TRANSFORMATION ALGORITHM

While the Engineering Vector contains values such as "Word size in decimal digits" and "Time to increment and test an index register," the equivalent Functional Vector has quantities such as "Number of fixed numeric input fields," "Number of simple update operations," "Number of characters in the master file," etc. To bring these elements into parallel is clearly going to take some reformatting, and some assumptions, such as exactly what constitutes a "Fixed Numeric Input Field."

These assumptions are made in the Engineering Transformation Algorithm and are fully documented in the algorithm itself. Typical entries (shown in Figure 10) deal with the actual performance which will be estimated for a magnetic tape unit when operating under known constraints. The algorithm sheets concerned are shown in Figure 11. It will be seen that the pre-requisite quantities are stated and evaluated on the following pages; the assumptions are then listed, the computation is stated, and the required quantities are formed.

The Engineering Transformation Algorithm is presented in an expository format which clearly explains the computational procedures and all necessary assumptions. Blanks are provided throughout to facilitate the working of example cases in order to gain familiarity with the procedures.

7. THE TRANSFORMED ENGINEERING VECTOR

This consists of a general background description of the computer, and four component "Vectors" which describe the performance of the system under three specific constraints and for the case of no known constraints. These are now in the correct units and form for use in the computation worksheets (in fact, the form of the Transformed Engineering Vector might well be a worksheet blank).

The quantities in the Transformed Engineering Vector have been given descriptive names as well as numbers. Thus, TEV 1 is the "Time taken to edit a fixed numeric field on input." These names, however, while correct in the restricted sense in which they are going to be used in the VECTOR process, should not be taken as being definitive for any wider application. (See Figure 12.)

Pre-computation of Magnetic TAPE PERFORMANCE ON DECIMAL DATA under CP Critical conditions.

<u>Pre-requisites:</u> PR 1901, PR 1902, PR 1915, PR 1916.

<u>Method</u>

This element can be arrived at by assuming that PR 1915 correctly estimates the block size to be used and PR 1916 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation, the value of the element concerned is:

((PR 1901) x (PR 1916)) x ((PR 1915) ÷ ((PR 1915) + (PR 1902)))

= ((   ) x (   )) x ((   ) ÷ ((   ) + (   )))

= ________________ digits per second.
(evaluated to 3 significant figures)

This gives a rate in digits/second. To transform this into the required microseconds/digits form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 19 (CP).

Figure 10. Specimen Page of Engineering Transformation Algorithm Showing Assumptions and Working Space

Pre-computation of working block size for decimal data when operating under CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1914, PR 1903, PR 1904.

Method

This element can be arrived at by assuming that the target block size, PR 1914, will be used where possible.

The quantities required are:

Minimum block size (PR 1903)
Maximum block size (PR 1904).

If (PR 1903) $\leq$ (PR 1914) and (PR 1904) $\geq$ (1914), then PR 1915 = PR 1914.

Otherwise, if (PR 1903) > (PR 1914), then PR 1915 = PR 1903; or if (PR 1904) < (PR 1914), then PR 1915 = PR 1904.

Since (PR 1903) = ________________,

(PR 1904) = ________________,

and (PR 1914) = ________________,

the value of PR 1915 = ________________.

Figure 11. Specimen Page of Engineering Transformation Algorithm Showing Partial Preparation of a Magnetic Tape Block Size

TRANSFORMED ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE UNDER GENERAL AND SPECIFIC LIMITING FACTORS			
		GEN	CP	I/O	SPACE
01	Edit a fixed numeric field during input				
02	Edit a fixed alphameric field during input				
03	Edit a flexible numeric field during input				
04	Edit a flexible alphameric field during input				
05	Simple Update Operation				
06	Complex Update				
07	Table Reference Time				
08	Edit a fixed numeric field during output				
09	Edit a fixed alphameric field during output				
10	Edit a flexible numeric field during output				
11	Edit a flexible alphameric field during output				
12	Control the processing of a record				
13	Control the movement of a record				
14	Load on central processor per alphameric character				
15	Load on central processor per decimal digit				
16	Load on central processor per card image				
17	Load on central processor per line image				
18	Magnetic tape performance on alphameric data				
19	Magnetic tape performance on decimal data				
20	Magnetic tape performance on card images				
21	Magnetic tape performance on line images				
22	Simultaneity rule number				
23	Simultaneity parameter number				

Figure 12. Specimen Page from Transformed Engineering Vector

ENGINEERING SPECIFICATION

SPECIFICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 201	x	x		x		Main memory size in words.	words	
ES 202		x		x		Word size in alphanumeric characters.	chars.	
ES 203		x		x		Word size in decimal digits.	digits	
ES 204	x					Word size in bits (excluding check bits).	bits	
ES 205					x	Main memory size in characters.	chars.	
ES 206			x			Main memory size in decimal digits.	digits	
ES 207			x			No. of decimal digits per alphanumeric character.		
ES 208	x	x	x	x	x	No. of index registers.		
ES 209	x	x		x		Main memory volume that can be accessed without indexing, in words.	words	
ES 210					x	Main memory volume that can be accessed without indexing, in characters.	chars.	
ES 211			x			Main memory volume that can be accessed without indexing, in decimal digits.	digits	

Abbreviations

Bin. FW = Binary Fixed Word Systems
Dec. FW = Decimal Fixed Word Systems
Dec. Char = Decimal Character Oriented Systems
AN FW = Alphanumeric Fixed Word Systems
AN Char = Alphanumeric Character Oriented Systems.

ENGINEERING SPECIFICATION

PART 3

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 301	x	x	x	x	x	Time taken to add two operands in main memory and store the sum (operands must have more than 4 decimal digits).	µsec.	
ES 302		x	x	x	x	Time taken to multiply an X digit operand by a Y digit operand and store the product (X and Y must be greater than 4).	µsec.	
ES 303		x	x	x	x	Time taken to divide an X digit operand by a Y digit operand and store the quotient (X and Y must be greater than 4).	µsec.	
ES 304	x					Time taken to multiply two operands in main memory and store the product.	µsec.	
ES 305	x					Time taken to divide two operands in main memory and store the quotient.	µsec.	
ES 306	x	x	x	x	x	Time taken to index in operand.	µsec.	
ES 307	x	x	x	x	x	Time taken to compare 2 operands in main memory (of at least 8 decimal digits or equivalent) and to transfer control to one of two arbitrary locations based on the result of the comparison.	µsec.	

Abbreviations

F = no. of character positions in operand field
N = no. of decimal digits in operand field

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 308	x	x	x	x	x	Time taken to perform the following task: A 1-digit operand, whose value is 1, 2, 3, 4, 5, or 6, is held in main memory. This is used to transfer control to one of six locations. The time stated includes a check that the value is between 1 and 6, and all necessary work up to and including the transfer of control. If the time varies, based on the value of the data item, quote a formula.	μsec.	
ES 309	x					Time taken for the following task: A four-bit operand is presently stored in the middle of a computer word. It is needed for use as a numeric operand, effectively right justified. The task is to prepare it for this use. (If no action need be taken, the time is zero.) Normally, it will be necessary to place it into another location.	μsec.	
ES 310		x		x		Time taken for the following task: A single-digit operand is presently stored in the middle of a computer word. It is needed for use as a numeric operand, effectivcly right justified. The task is to prepare it for this use. (If no action need bc taken, the time is zero.) Normally, it will be necessary to place it into another location.	μsec.	

Abbreviations

ENGINEERING SPECIFICATION

PART 3 (Cont'd)

SPECIFICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 311	x					Time taken for the following task, which is the complement of ES 309: A 4-bit operand has been produced by arithmetic operations and stored for continued computational use. What is the time needed to store it in the middle of a computer word without changing the contents of the rest of the word? (If ES 309 was zero, probably this will be zero also.)	μsec.	
ES 312		x		x		Time taken for the following task, which is the complement of ES 310: A one-digit operand has been produced by arithmetic operations and stored for continued computational use. What is the time needed to store it in the middle of a computer word without changing the contents of the rest of the word? (If ES 310 was zero, probably this will be zero also.)	μsec.	
ES 313	x	x	x	x	x	Time taken to increment and test an index register.	μsec.	
ES 314	x	x	x	x	x	Time taken to move an instruction from one part of the main memory to another location.	μsec.	

Abbreviations

PART 3 (Cont'd)

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 315	x	x	x	x	x	Time taken to move a record of N characters from one part of the main memory to another. (N is to be considered a large number.)	μsec.	
ES 316	x	x	x	x	x	Time taken to indirectly address an operand. (If there is no indirect addressing capability, enter "∞".)	μsec.	

Abbreviations

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 401	x	x		x		General input editing task* with programming minimized and 11-character alphabetic field. Input field is synchronized (i.e., aligned in accordance with computer word structure).	μsec.	
ES 402	x	x		x		General input editing task* with programming minimized and 5-digit numeric field. Input field is synchronized.	μsec.	
ES 403	x	x		x		General input editing task* with object time minimized and 11-character alphabetic field. Input field is synchronized.	μsec.	
ES 404	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is synchronized.	μsec.	
ES 405	x	x		x		General input editing task* with programming minimized, and 11-character alphabetic field. Input field is not synchronized (i.e., it overlaps computer word boundaries).	μsec.	

* General input editing task: Take a field stored in main memory in punched card code; verify the legality of the punching; translate as needed; and unpack so that the field can be used directly as an arithmetic operand. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic or numeric fields; and (for fixed word systems only) input fields synchronized or not synchronized with the computer's word structure. (Where radix conversion is required between card code and computational representation, the conversion time should be included unless the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 406	x	x		x		General input editing task* with programming minimized, and 5-digit numeric field. Input field is not synchronized.	μsec.	
ES 407	x	x		x		General input editing task* with object time minimized and 11-character alphabetic field. Input field is not synchronized.	μsec.	
ES 408	x	x		x		General input editing task* with object time minimized and 5-digit numeric field. Input field is not synchronized.	μsec.	
ES 409			x		x	General input editing task* with programming minimized and 11-character alphabetic field.	μsec.	
ES 410			x		x	General input editing task* with programming minimized and 5-digit numeric field.	μsec.	

Abreviations

ENGINEERING SPECIFICATION

PART 4 (Cont'd)

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 411			x		x	General input editing task* with object time minimized and 11-character alphabetic field.	μsec.	
ES 412			x		x	General input editing task* with object time minimized and 5-digit numeric field.	μsec.	

Abbreviations

SPECIFICATION NO.	APPLICABLE FOR Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.	QUERY	ANSWER	COMMENT NO.
ES 413	x	x		x		General output editing task* with programming minimized and an 11-character alphabetic field. Output field is synchronized.	µsec.	
ES 414	x	x		x		General output editing task* with programming minimized and a 6-character numeric field and commercial editing. Output field is synchronized.	µsec.	
ES 415	x	x		x		General output editing task* with programming minimized and a 6-character numeric field and scientific editing. Output field is synchronized.	µsec.	
ES 416	x	x		x		General output editing task* with object time minimized and an 11-character alphabetic field. Output field is synchronized.	µsec.	
ES 417	x	x		x		General output editing task* with object time minimized and a 6-character numeric field and commercial editing. Output field is synchronized.	µsec.	

* General output editing task: Take a field stored in main memory, insert editing symbols, translate to printer code as needed, and move an output area in main memory. The times are differentiated into coding with minimized programming effort or minimized object time; alphabetic, commercial numeric, or scientific numeric fields (see below); and (for fixed word systems only) output fields synchronized or not synchronized with the computer's word structure.

- Alphabetic field: The stored field is simply moved to the output area, with translation to printer code if needed.

- Commercial editing on numeric field: The stored field is in cents. Insert floating dollar sign, comma, and decimal point. Place CR or DB alongside, depending upon the sign.

- Scientific editing on numeric field: The stored field requires zero suppression and insertion of a sign and decimal point, with two decimal places to the right of the point.

(Where numeric fields require radix conversion between the computational representation and the printer code, the conversion time should be included unless the radix conversion can be more efficiently performed off-line. In the latter case, please describe the equipment and procedure to be used for the off-line radix conversion.)

SPECIFICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 418	x	x		x		General output editing task* with object time minimized and a 6-character numeric field and scientific editing. Output field is synchronized.	μsec.	
ES 419	x	x		x		General output editing task* with programming minimized and an 11-character alphabetic field. Output field is not synchronized.	μsec.	
ES 420	x	x		x		General output editing task* with programming minimized and a 6-character numeric field and commercial editing. Output field is not synchronized.	μsec.	
ES 421	x	x		x		General output editing task* with programming minimized and 6-character numeric field and scientific field and scientific editing. Output field is not synchronized.	μsec.	
ES 422	x	x		x		General output editing task* with object time minimized and an 11-character alphabetic field. Output field is not synchronized.	μsec.	

Abbreviations

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 423	x	x		x		General output editing task* with object time minimized and a 6-character numeric field and commerical editing. Output field is not synchronized.	μsec.	
ES 424	x	x		x		General output editing task* with object time minimized and a 6-character numeric field and scientific editing. Output field is not synchronized.	μsec.	
ES 425			x		x	General output editing task* with programming minimized and an 11-character alphabetic field.	μsec.	
ES 426			x		x	General output editing task* with programming minimized and a 6-character numeric field and commercial editing.	μsec.	
ES 427			x		x	General output editing task* with programming minimized and a 6-character numeric field and scientific editing	μsec.	

Abbreviations

ENGINEERING SPECIFICATION

PART 4 (Cont'd)

SPECIFICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 428			x		x	General output editing task* with object time minimized and an 11-character alphabetic field.	μsec.	
ES 429			x		x	General output editing task* with object time minimized and a 6-character numeric field and commercial editing.	μsec.	
ES 430			x		x	General output editing task* with object time minimized and a 6-character numeric field and scientific editing.	μsec.	

Abbreviations

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 450	x	x	x	x	x	General table search task* with minimized programming, using sequential search.	μsec.	
ES 451	x	x	x	x	x	General table search task* with minimized object time, using sequential search method.	μsec.	
ES 452	x	x	x	x	x	General table search task* with minimized programming, using binary search method.	μsec.	
ES 453	x	x	x	x	x	General table search task* with minimized object time, using binary search method.	μsec.	

* General table search task: Examine a table stored in main memory to find an argument identical with a test argument. The desired answer is the time per argument tested, with initialization time ignored. Arguments are 8 decimal digits long, and arranged in ascending sequence with variable increments between the values of consecutive arguments.

- Sequential search method: The table arguments are examined in straightforward sequential fashion, allowing the automatic table look-up facilities to be used in many computers.

- Binary search method: Assumc the table has N entries, where N is 2 raised to any integral power (e.g., 64). First compare the (N/2)th table argument with the test argument. Depending upon the results, examinc next the (N/4)th or (3N/4)th argument; then the (N/8)th, (3N/8)th, (5N/8)th, or (7N/8)th argument; etc.

Abbreviations

ENGINEERING SPECIFICATION

PART 5

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 501	x	x	x	x	x	Number of magnetic tapes which can be reading with processing proceeding.		
ES 502	x	x	x	x	x	Number of magnetic tapes which can be reading with no processing proceeding.		
ES 503	x	x	x	x	x	Number of magnetic tapes which can be writing with processing proceeding.		
ES 504	x	x	x	x	x	Number of magnetic tapes which can be writing with no processing proceeding.		
ES 505	x	x	x	x	x	Total number of magnetic tapes which can be reading and/or writing with processing proceeding.		
ES 506	x	x	x	x	x	Total number of magnetic tapes which can be reading and/or writing with no processing proceeding.		
ES 507	x	x	x	x	x	Can more than one program be running at one time? (Yes or No)		

Abbreviations

ENGINEERING SPECIFICATION

PART 6

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 601	x	x	x	x	x	Peak speed, in alphanumeric characters per second.	char/sec.	
ES 602	x	x	x	x	x	Cost in characters of a tape gap when passed over as quickly as possible. *	chars.	
ES 603	x	x	x	x	x	Minimum block length, in alphanumeric characters.	chars.	
ES 604	x	x	x	x	x	Block length increment, in alpha-numeric characters.	chars.	
ES 605	x	x	x	x	x	Maximum block length, in alpha-numeric characters.	chars.	
ES 606	x	x	x	x	x	Central processor time used per alphanumeric character read or written.	μsec.	
ES 607	x	x	x	x	x	Number of decimal digits per alpha-numeric character in the computer's internal code.		

* Can be determined by multiplying the minimum time to cross the inter-block gap, in seconds, by the peak data transfer rate, in characters per second.

Abbreviations

SPECI-FICATION NO.	APPLICABLE FOR					QUERY	ANSWER	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
ES 608	x	x	x	x	x	Number of decimal digits per alpha-numeric character in the magnetic tape code.		
ES 609	x	x	x	x	x	Number of alphanumeric characters per computer word.		
ES 610	x	x	x	x	x	Maximum blocking factor for card image input available using standard routines.		
ES 611	x	x	x	x	x	Maximum blocking factor for line images output available using standard routines.		
ES 612	x	x	x	x	x	Additional central processor time used per alphanumeric character when scatter-read gather-write facilities are used. (If such facilities are not avail-able, write N. A.).	μsec.	

Abbreviations

ENGINEERING CONVERSION ALGORITHM

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT*	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1201 Memory size in alphanumeric characters	x					(ES 201) x (ES 204) ÷ 6		
		x				(ES 201) x (ES 202)		
			x			(ES 206) ÷ (ES 207)		
				x		(ES 201) x (ES 202)		
					x	(ES 205)		
EV 1202 Memory size in decimal digits	x					(ES 201) x (ES 204) x 0.3		
		x				(ES 201) x (ES 203)		
			x			(ES 206)		
				x		(ES 201) x (ES 203)		
					x	(ES 205)		
EV 1203 Word size in alphanumeric characters	x					(ES 204 ÷ 6), to 2 decimal places		
		x				(ES 202)		
			x			Reciprocal of (ES 207)		
				x		(ES 202)		
					x	1.0		
EV 1204 Word size in decimal digits	x					(ES 204) x 0.3		
		x				(ES 203)		
			x			1.0		
				x		(ES 203)		
					x	1.0		
EV 1205 Number of index registers	x	x	x	x	x	(ES 208)		
EV 1206 Volume of memory (in alphanumeric characters) that can be accessed without indexing	x					(ES 209) x (ES 204) ÷ 6		
		x				(ES 209) x (ES 202)		
			x			(ES 211) ÷ (ES 207)		
				x		(ES 209) x (ES 202)		
					x	(ES 210)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instruction.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT * μsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1301 Addition time	x	x	x	x	x	(ES 301), evaluating any formula with: No. of digits = 5. No. of significant digits = 5.		
EV 1302 Multiplication time	x					(ES 304), evaluating any formula with: No. of "1" bits in either operand = 8.		
		x	x	x	x	(ES 304), evaluating any formula with: No. of digits in either operand = 5. Sum of digits in either operand = 25. Complement of digits in either operand = 20.		
EV 1303 Division time	x					(ES 303), evaluating any formula with: No. of "1" bits in either operand = 8.		
		x	x	x	x	(ES 304), evaluating any formula with: No. of digits in either operand = 5. Sum of digits in either operand = 25. Complement of digits in either operand = 20.		
EV 1304 Indexing time	x	x	x	x	x	(ES 306)		
EV 1305 Time to increment and test an index register	x	x	x	x	x	(ES 313)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT* µsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1306 Indirect addressing time	x	x	x	x	x	(ES 316), if no answer, enter "∞"		
EV 1307 Unpacking time	x					(ES 309)		
		x				(ES 310), evaluating any formula with given digit = 4.		
			x			0.0		
				x		(ES 310), evaluating any formula with given digit = 4.		
					x	0.0		
EV 1308 Packing time	x					(ES 311)		
		x				(ES 312), evaluating any formula with given digit = 4.		
			x			0.0		
				x		(ES 312), evaluating any formula with given digit = 4.		
					x	0.0		
EV 1309 Instruction move time	x	x	x	x	x	(ES 314)		
EV 1310 Character move time	x	x	x	x	x	(ES 315), evaluated with N = 1; disregard all terms except the one containing N.		
EV 1311 Single comparison time	x	x	x	x	x	(ES 307)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT * µsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Chor.	AN FW	AN Chor.			
EV 1401 Input edit: programming minimized, alphanumeric synchronized	x	x	x	x	x	(ES 401) (ES 409)		
EV 1402 Input edit: programming minimized, numeric, synchronized	x	x	x	x	x	(ES 402) (ES 410)		
EV 1403 Input edit: object time minimized, alphanumeric, synchronized	x	x	x	x	x	(ES 403) (ES 411)		
EV 1404 Input edit: object time minimized, numeric, synchronized	x	x	x	x	x	(ES 404) (ES 412)		
EV 1405 Input edit: programming minimized, alphanumeric, not synchronized	x	x	x	x	x	(ES 405) (ES 409)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT* μsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1406 Input edit: programming minimized, numeric, not synchronized	x	x	x	x	x	(ES 406) (ES 410)		
EV 1407 Input edit: object time minimized, alphanumeric, not synchronized	x	x	x	x	x	(ES 407) (ES 411)		
EV 1408 Input edit: object time minimized, numeric, not synchronized	x	x	x	x	x	(ES 408) (ES 412)		
EV 1409 Output edit: programming minimized, alphanumeric, synchronized	x	x	x	x	x	(ES 413) (ES 425)		
EV 1410 Output edit: programming minimized, commercial, synchronized	x	x	x	x	x	(ES 415) (ES 427)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT* μsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1411 Output edit: programming minimized, scientific, synchronized	x	x	x	x	x	(ES 415) (ES 428)		
EV 1412 Output edit: object time minimized, alphanumeric, synchronized	x	x	x	x	x	(ES 416) (ES 428)		
EV 1413 Output edit: object time minimized, commercial, synchronized	x	x	x	x	x	(ES 417) (ES 429)		
EV 1414 Output edit: object time minimized, scientific, synchronized	x	x	x	x	x	(ES 418) (ES 430)		
EV 1415 Output edit: programming minimized, alphanumeric, not synchronized	x	x	x	x	x	(ES 419) (ES 425)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT* μsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1416 Output edit: programming minimized, commercial, not synchronized	x	x	x	x	x	(ES 420) (ES 426)		
EV 1417 Output edit: programming minimized, scientific, not synchronized	x	x	x	x	x	(ES 421) (ES 427)		
EV 1418 Output edit: object time minimized, alphanumeric, not synchronized	x	x	x	x	x	(ES 422) (ES 428)		
EV 1419 Output edit: object time minimized, commercial, not synchronized	x	x	x	x	x	(ES 423) (ES 429)		
EV 1420 Output edit: object time minimized, scientific, not synchronized	x	x	x	x	x	(ES 424) (ES 430)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT * µsec.	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1450 Sequential comparison step; programming minimized	x	x	x	x	x	(ES 450), evaluating any formula with number of arguments to bc searched prior to a find equalling 1, and neglecting all other terms.		
EV 1451 Sequential comparison step; object time minimized	x	x	x	x	x	(ES 451), evaluating any formula as for EV 1450, above		
EV 1452 Binary comparison step; programming minimized	x	x	x	x	x	(ES 452), evaluating any formula for a table size of 2 entries and neglecting all other terms.		
EV 1453 Binary comparison step; object time minimized	x	x	x	x	x	(ES 453), evaluating any formula as for EV 1452, above.		

* Computed to one place to right of decimal point unless otherwisc stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT *	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1501 Tape reads with processing	x	x	x	x	x	(ES 501)		
EV 1502 Tape reads without processing	x	x	x	x	x	(ES 502)		
EV 1503 Tape writes with processing	x	x	x	x	x	(ES 503)		
EV 1504 Tape writes without processing	x	x	x	x	x	(ES 504)		
EV 1505 Tape transfers (read plus write) with processing	x	x	x	x	x	(ES 505)		
EV 1506 Tape transfers (read plus write) without processing	x	x	x	x	x	(ES 506)		
EV 1507 Multirunning possibilities	x	x	x	x	x	If (ES 507) is "No", enter 0. If (ES 507) is "Yes", enter 1.		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT *	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1601 Peak speed in alphanumeric characters per second	x	x	x	x	x	(ES 601)		
EV 1602 Minimum gap cost in alpha-numeric characters	x	x	x	x	x	(ES 602)		
EV 1603 Minimum block length in alphanumeric characters	x	x	x	x	x	(ES 603)		
EV 1604 Block length increment in alphanumeric characters	x	x	x	x	x	(ES 604)		
EV 1605 Maximum block length in alphanumeric characters	x	x	x	x	x	(ES 605)		
EV 1606 Central processor time used per alphanumeric character read or written, in microseconds	x	x	x	x	x	(ES 606)		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR ELEMENT	APPLICABLE FOR					INSTRUCTIONS	RESULT *	COMMENT NO.
	Bin. FW	Dec. FW	Dec. Char.	AN FW	AN Char.			
EV 1607 Decimal digits per alphanumeric character in internal representation	x	x	x	x	x	(ES 607)		
EV 1608 Decimal digits per alphanumeric character on magnetic tape	x	x	x	x	x	(ES 608)		
EV 1609 Number of alphanumeric characters per computer word	x	x	x	x	x	(ES 609)		
EV 1610 Maximum blocking factor for card image input using standard routines	x	x	x	x	x	(ES 610)		
EV 1611 Maximum blocking factor for line image output using standard routines	x	x	x	x	x	(ES 611)		
EV 1612 Additional time per character when scatter read or gather write facilities are used	x	x	x	x	x	(ES 612), replacing "N. A." (if used) with "00"		

* Computed to one place to right of decimal point unless otherwise stipulated in the instructions.

ENGINEERING VECTOR

ENGINEERING VECTOR

ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE
1201	Memory size in alphanumeric characters	
1202	Memory size in decimal digits	
1203	Word size in alphanumeric characters	
1204	Word size in decimal digits	
1205	Number of index registers	
1206	Volume of memory (in alphanumeric characters) that can be accessed without indexing	
1301	Addition time	
1302	Multiplication time	
1303	Division time	
1304	Indexing time	
1305	Time to increment and test an index register	
1306	Indirect addressing time	
1307	Unpacking time	
1308	Packing time	
1309	Instruction move time	
1310	Character move time	
1311	Single comparison time	
1401	Input edit: programming minimized, alphanumeric, synchronized	
1402	Input edit: programming minimized, numeric, synchronized	
1403	Input edit: object time minimized, alphanumeric, synchronized	
1404	Input edit: object time minimized, numeric, synchronized	
1405	Input edit: programming minimized, alphanumeric, not synchronized	

ENGINEERING VECTOR (Cont'd)

ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE
1406	Input edit: programming minimized, numeric not synchronized	
1407	Input edit: object time minimized, alpha-numeric, not synchronized	
1408	Input edit: object time minimized, numeric not synchronized	
1409	Output edit: programming minimized, alphanumeric, synchronized	
1410	Output edit: programming minimized, commercial, synchronized	
1411	Output edit: programming minimized, scientific, synchronized	
1412	Output edit: object time minimized, alphanumeric, synchronized	
1413	Output edit: object time minimized, commercial, synchronized	
1414	Output edit: object time minimized, scientific, synchronized	
1415	Output edit: programming minimized, alphanumeric, not synchronized	
1416	Output edit: programming minimized, commercial, not synchronized	
1417	Output edit: programming minimized, scientific, not synchronized	
1418	Output edit: object time minimized, alphanumeric, not synchronized	
1419	Output edit: object time minimized, commercial, not synchronized	
1420	Output edit: object time minimized, scientific, not synchronized	

ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE
1450	Sequential comparison step; programming minimized	
1451	Sequential comparison step; object time minimized	
1452	Binary comparison step; programming minimized	
1453	Binary comparison step; object time minimized	
1501	Tape reads with processing	
1502	Tape reads without processing	
1503	Tape writes with processing	
1504	Tape writes without processing	
1505	Tape transfers (read plus write) with processing	
1506	Tape transfers (read plus write) without processing	
1507	Multirunning possibilities	
1601	Peak speed in alphanumeric characters per second	
1602	Minimum gap cost in alphanumeric characters	
1603	Minimum block length in alphanumeric characters	
1604	Block length increment in alphanumeric characters	
1605	Maximum block length in alphanumeric characters	
1606	Central processor time used per alpha-numeric character read or written in microseconds	

ENGINEERING VECTOR (Cont'd)

ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE
1607	Decimal digits per alphanumeric character in internal representation	
1608	Decimal digits per alphanumeric character on magnetic tape	
1609	Number of alphanumeric characters per computer word	
1610	Maximum blocking factor for card image input using standard routines	
1611	Maximum blocking factor for line image output using standard routines	
1612	Additional time per character when scatter read or gather write facilities are used	

ENGINEERING TRANSFORMATION ALGORITHM

Pre-computation of the time taken to EDIT A FIXED NUMERIC FIELD DURING INPUT under CP Critical conditions.

Pre-requisites: PR 0101.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,
2. Synchronization will be correctly computed in PR 0101,

and then computing the time involved when running time is minimized in:

(Input editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Input editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0101), or are included in the Engineering Vector (EV 1404, EV 1408), the value of TEV 01 (CP) is:

(EV 1404) x (PR 0101) + (EV 1408) x (1 - (PR 0101))

= (_______) x (_______) + (_______) x (____________)

= _________ microseconds.

Pre-computation of synchronization of numeric data under CP Critical conditions.

Pre-requisites: None

Method

This element can be arrived at by assuming that if the word size is 1, 2, or 3 digits, complete synchronization will be used; otherwise, the proportion of fields synchronized will be 3/(word length in decimal digits).

As this value is included in the Engineering Vector (EV 1204), the value of PR 0101 is:

1 if EV 1204 is 1, 2, or 3;
or 3/(EV 1204) if EV 1204 is greater than 3;

i.e., as EV 1204 = __________,

PR 0101 = __________________.

Pre-computation of the time taken to EDIT A FIXED NUMERIC FIELD DURING INPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: PR 0102.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,
2. Synchronization will be correctly computed in PR 0102,

and then computing the time involved when programming is minimized in:

(Input editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Input editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0102), or are included in the Engineering Vector (EV 1402, EV 1406), the value of TEV 01 (G, IO, S) is:

(EV 1402) x (PR 0102) + (EV 1406) x (1 - (PR 0102))

= (_______) x (_______) + (_______) x (___________)

= __________ microseconds.

Pre-computation of synchronization of numeric data under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that synchronization will be inversely proportional to the word length in numeric digits.

As this value is included in the Engineering Vector (EV 1204), the value of PR 0102 is:

1/(EV 1204); i.e., 1/(________) = _________.

Pre-computation of the time taken to EDIT A FIXED ALPHAMERIC FIELD DURING INPUT under CP Critical conditions.

Pre-requisites: PR 0201.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,
2. Synchronization will be correctly computed in PR 0201,
3. Fields not presently known to be numeric are still equally likely to require numeric handling,

and then computing the following: One-half of the time involved when running time is minimized in:

> (Input editing a synchronized alphameric field + Input editing a synchronized numeric field) x (Proportion of synchronized fields) + (Input editing a non-synchronized alphameric field + Input editing a non-synchronized numeric field) x (Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0201), or are included in the Engineering Vector (EV 1403, EV 1404, EV 1407, EV 1408), the value of TEV 02 (CP) is:

1/2((EV 1403) + (EV 1404)) x (PR 0201) + 1/2((EV 1407) + (EV 1408)) x (1 - (PR 0201))

= 1/2((_______) + (_______)) x (_______) + 1/2((_______) + (_______)) x (_________)

= ______________ microseconds.

Pre-computation of synchronization of alphameric data under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that if the word size is 1, 2, or 3 characters, complete synchronization will be used; otherwise, the proportion of synchronized fields will be 3/(word length in alphameric characters).

As this value is included in the Engineering Vector (EV 1203), the value of PR 0201 is:

1 if EV 1203 is 1, 2, or 3;
or 3/(EV 1203) if EV 1203 is greater than 3;

i.e., as EV 1203 = ______________,

PR 0201 = ______________.

Pre-computation of the time taken to EDIT A FIXED ALPHAMERIC FIELD DURING INPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: PR 0202.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,

2. Synchronization will be correctly computed in PR 0202,

3. Fields not presently known to be numeric are still equally likely to require numeric handling,

and then computing the following: One-half of the time involved when programming is minimized in:

(Input editing a synchronized alphameric field + Input editing a synchronized numeric field) x (Proportion of synchronized fields) + (Input editing a non-synchronized alphameric field + Input editing a non-synchronized numeric field) x (Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0202), or are included in the Engineering Vector (EV 1401, EV 1402, EV 1405, EV 1406), the value of the element concerned is:

1/2((EV 1401) + (EV 1402)) x (PR 0202) + 1/2((EV 1405) + (EV 1406)) x (1 - (PR 0202))

= 1/2((_______) + (_______)) x (_______) + 1/2((_______) + (_______)) x (_________))

= ____________ microseconds.

Pre-computation of synchronization of alphameric data under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that synchronization will be inversely proportional to the word length in alphameric characters.

As this value is included in the Engineering Vector (EV 1203), the value of PR 0202 is:

1/(EV 1203); i.e., 1/(__________) = ____________.

Pre-computation of the time taken to EDIT A FLEXIBLE NUMERIC FIELD DURING INPUT under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,

2. Synchronization will be complete,

and then noting the time involved when running time is minimized in input editing a synchronized numeric field.

As this value is included in the Engineering Vector (EV 1404), the value of TEV 03 (CP) is equal to EV 1404, or ____________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE NUMERIC FIELD DURING INPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,

2. Synchronization will be complete,

and then noting the time involved when programming is minimized in input editing a synchronized numeric field.

As this value is included in the Engineering Vector (EV 1402), the value of TEV 03 (G, IO, S) is equal to EV 1402, or __________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE ALPHAMERIC FIELD DURING INPUT under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,
2. Synchronization will be complete,
3. Fields not presently known to be numeric are still equally likely to require numeric handling,

and then computing the following: One-half of the time involved when running time is minimized in:

(Input editing a synchronized alphameric field +
Input editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1403, EV 1404), the value of TEV 04 (CP) is:

1/2((EV 1403) + (EV 1404))

= 1/2((________) + (________))

= ____________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE ALPHAMERIC FIELD DURING INPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,

2. Synchronization will be complete,

3. Fields not presently known to be numeric are still equally likely to require numeric handling,

and then computing the following: One-half of the time involved when programming is minimized in:

(Input editing a synchronized alphameric field +
Input editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1401, EV 1402), the value of TEV 04 (G, IO, S) is:

1/2((EV 1401) + (EV 1402))

= 1/2((________) + (________))

= ______________ microseconds.

Pre-computation of time taken to execute a SIMPLE UPDATE OPERATION under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that for each known, functional addition, subtraction, or comparison, a computer system will execute 5 equivalent operations, and then computing the following:

2 x (time taken to add two operands and store the result) +
2 x (time taken to compare two quantities and branch) +
(time taken to move one instruction).

As these values are included in the Engineering Vector (EV 1301, EV 1311, EV 1309), the value of TEV 05 (CP) is:

2 x (EV 1301) + 2 x (EV 1311) + (EV 1309)

= 2 x (________) + 2 x (________) + (________)

= ______________microseconds.

Pre-computation of time taken to execute a SIMPLE UPDATE OPERATION under General/ I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the required time will vary from the time taken to perform this task under CP Critical conditions in the following ways:

1. The average operand will be indexed or indirectly addressed,

2. Loop control will be needed once every 10 references to an operand. Thus, for each operand (of which there are 10), 1 address modification (the lesser of EV 1304 or EV 1306) and 1/10 of a "step and test" (EV 1305) will be needed.

As these values are included in the Engineering Vector (EV 1301, EV 1311, EV 1309, EV 1304, EV 1306, EV 1305), the value of TEV 05 (G, IO) is:

2 x (EV 1301) + 2 x (EV 1311) + (EV 1309) + 10 x (the lesser of EV 1304 or EV 1306) + (EV 1305)

= 2 x (_______) + 2 x (_______) + (_______) + 10 x (_______) + (_______)

= __________ microseconds.

Pre-computation of time taken to execute a SIMPLE UPDATE OPERATION under Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the required time will vary from the time taken to perform this task under CP Critical conditions in the following ways:

1. The average operand will be indexed or indirectly addressed,

2. Loop control will be needed once every 4 references to an operand. Thus, for each operand (of which there are 10), 1 address modification (the lesser of EV 1304 or EV 1306) and 1/4 of a "step and test" (EV 1305) will be needed.

As these values are included in the Engineering Vector (EV 1301 EV 1311, EV 1309, EV 1304, EV 1306, EV 1305), the value of TEV 05 (S) is:

2 x (EV 1301) + 2 x (EV 1311) + (EV 1309) + 10 x ((the lesser of EV 1304 or EV 1306) + 1/4 x (EV 1305))

= 2 x (_______) + 2 x (_______) + (_______) + 10 x ((_______) + 1/4 x (_______))

= ____________ microseconds.

Pre-computation of time taken to execute a COMPLEX UPDATE STEP under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that for each known, functional multiplication or division, a computer system will execute 1 multiplication, 3 additions, 2 comparisons, and 3 moves; and then computing the following:

(Multiply time) + 3 x (addition time) + 2 x (comparison time) +
3 x (instruction move time).

As these values are included in the Engineering Vector (EV 1302, EV 1301, EV 1311, EV 1309), the value of TEV 06 (CP) is:

(EV 1302) + 3 x (EV 1301) + 2 x (EV 1311) + 3 x (EV 1309)

= (_______) + 3 x (_______) + 2 x (_______) + 3 x (_______)

= __________ microseconds.

Pre-computation of time taken to execute a COMPLEX UPDATE STEP under General/ I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the required time will vary from the time taken to perform this task under CP Critical conditions in the following way:

1. The average operand will be indexed or indirectly addressed,

2. Loop control will be needed once every 10 references to an operand. Thus, for each operand (of which there are 16), 1 modification (the lesser of EV 1304 or EV 1306) and 1/10 of a "step and test" (EV 1305) will be needed.

As these values are included in the Engineering Vector (EV 1302, EV 1301, EV 1311, EV 1309, EV 1304, EV 1306, EV 1305), the value of TEV 06 (G, IO) is:

(EV 1302) + 3 x (EV 1301) + 2 x (EV 1311) + 3 x (EV 1309) + 16 x ((the lesser of EV 1304 or EV 1306) + 1/10 x (EV 1305))

= (_______) + 3 x (_______) + 2 x (_______) + 3 x (_______) + 16 x ((______) + 1/10 x (______))

= ____________ microseconds.

Pre-computation of time taken to execute a COMPLEX UPDATE STEP under Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the required time will vary from the time taken to perform this task under CP Critical conditions in the following ways:

1. The average operand will be indexed or indirectly addressed,

2. Loop control will be needed once every 4 references to an operand. Thus, for each operand (of which there are 16), 1 modification (the lesser of EV 1304 or EV 1306) and 1/4 of a "step and test" (EV 1305) will be needed.

As these values are included in the Engineering Vector (EV 1302, EV 1301, EV 1311, EV 1309, EV 1304, EV 1306, EV 1305), the value of TEV 06 (S) is:

(EV 1302) + 3 x (EV 1301) + 2 x (EV 1311) + 3 x (EV 1309) +
16 x ((the lesser of EV 1304 or EV 1306) + 1/4 x (EV 1305))

= (_______) + 3 x (_______) + 2 x (_______) + 3 x (_______) +
16 x ((_______) + 1/4 x (_______))

= __________ microseconds.

Pre-computation of TABLE REFERENCE TIME per essential binary step under CP Critical conditions.

<u>Pre-requisites</u>: None.

<u>Method</u>

This element can be arrived at by assuming that either the binary search or sequential search method will be used; and that it will be possible to replace any single binary search operation by searching sequentially through a 10-value table. Thus the lesser of the following two times should be used: binary comparison step or 5 x (sequential comparison step).

As these values are included in the Engineering Vector (EV 1451, EV 1453), the value of TEV 07 (CP) is:

The lesser of (EV 1453) or (5 x (EV 1451));

i.e., the lesser of (______) or (5 x (______))

= ____________ microseconds.

Pre-computation of TABLE REFERENCE TIME per essential binary step under General/ I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that this will vary from the time taken to perform this task under CP Critical conditions in that programming rather than running time will be minimized; i.e., EV 1450 and EV 1452 will be substituted for EV 1451 and EV 1453 in the CP Critical computation.

Therefore, the value of TEV 07 (G, IO, S) is:

The lesser of (EV 1452) or (5 x (EV 1450));

i.e., the lesser of (_______) or (5 x (_______))

= ___________ microseconds.

Pre-computation of the time taken to EDIT A FIXED NUMERIC FIELD DURING OUTPUT under CP Critical conditions.

Pre-requisites: PR 0801.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,
2. Synchronization will be correctly computed in PR 0801,
3. Output fields are equally likely to require scientific or commercial editing,

and then computing the following: One-half of the time involved when running time is minimized in:

(Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Commercial output editing a non-synchronized numeric field +
Scientific output editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0801), or are included in the engineering vector (EV 1413, EV 1414, EV 1419, EV 1420), the value of the element concerned is:

1/2((EV 1413) + (EV 1414)) x (PR 0801) + 1/2((EV 1419) + (EV 1420)) x (1 - (PR 0801))

= 1/2((______) + (______)) x (______) + 1/2((______) + (______)) x (______)

= __________ microseconds.

Pre-computation of synchronization of numeric data under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that if the word size is 1, 2, or 3 digits, complete synchronization will be used; otherwise, the proportion of fields synchronized will be 3/(word length in decimal digits).

As this value is included in the Engineering Vector (EV 1204), the value of PR 0801 is:

1 if EV 1204 is 1, 2, 3;
or 3/(EV 1204) if EV 1204 is greater than 3;

i.e., as EV 1204 = ____________,

PR 0801 = ____________.

Pre-computation of the time taken to EDIT A FIXED NUMERIC FIELD DURING OUTPUT under General/I-O Critical/Space Critical conditions.

Pre-requisistes: PR 0802.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,
2. Synchronization will be correctly computed in PR 0802,
3. Output fields are equally likely to require scientific or commercial editing,

and then computing the following: One-half of the time involved when programming is minimized in:

(Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Commercial output editing a non-synchronized numeric field +
Scientific output editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0802), or are included in the Engineering Vector (EV 1410, EV 1411, EV 1416, EV 1417), the value of the element concerned is:

1/2((EV 1410) + (EV 1411)) x (PR 0802) + 1/2((EV 1416) + (EV 1417)) x (1 - (PR 0802))

= 1/2((________) + (________)) x (________) + 1/2((________) + (________)) x (__________)

= ____________ microseconds.

Pre-computation of synchronization of numeric data under General/I-O Critical/Space Critical conditions.

Pre-requisities: None.

Method

This element can be arrived at by assuming that synchronization will be inversely proportional to the word length in numeric digits.

As this value is included in the Engineering Vector (EV 1204), the value of PR 0802 is:

1/(EV 1204); i.e., 1/(________) = _________.

Pre-computation of the time taken to EDIT A FIXED ALPHAMERIC FIELD DURING OUTPUT under CP Critical conditions.

Pre-requisites: PR 0901.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,
2. Synchronization will be correctly computed in PR 0901,
3. Fields not presently known to be numeric are still equally likely to require numeric handling,
4. Output fields are equally likely to require alphabetic, scientific, or commercial editing,

and then computing the following: One-third of the time involved when running time is minimized in:

(Output editing a synchronized alphameric field +
Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Output editing a non-synchronized alphameric field +
Commercial output editing a non-synchronized numeric field +
Scientific output editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0901), or are included in the Engineering Vector (EV 1412, EV 1413, EV 1414, EV 1418, EV 1419, EV 1420), the value of TEV 09 (CP) is:

1/3((EV 1412) + (EV 1413) + (EV 1414)) x (PR 0901) +

1/3((EV 1418) + (EV 1419) + (EV 1420)) x (1 - (PR 0901))

= 1/3((_______) + (_______) + (_______)) x (_______) +

1/3((_______) + (_______) + (_______)) x (__________))

= __________ microseconds.

Pre-computation of synchronization of alphameric fields under CP Critical conditions.

Pre-requisites: None

Method

This element can be arrived at by assuming that if the word size is 1, 2, or 3 characters, synchronization will be complete; otherwise, the proportion of synchronized fields will be 3/(word length in alphameric characters).

As this value is included in the Engineering Vector (EV 1203), the value of PR 0901 is:

1 if EV 1203 is 1, 2, or 3;
or 3/(EV 1203) if EV 1203 is greater than 3;

i.e., as EV 1203 = ________,

PR 0901 = ________.

Pre-computation of the time taken to EDIT A FIXED ALPHAMERIC FIELD DURING OUTPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: PR 0902.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,

2. Synchronization will be correctly computed in PR 0902,

3. Fields not presently known to be numeric are still equally likely to require numeric handling,

4. Output fields are equally likely to require alphabetic, scientific, or commercial editing,

and then computing the following: One-third of the time involved when programming is minimized in:

(Output editing a synchronized alphameric field +
Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field) x
(Proportion of synchronized fields) +
(Output editing a non-synchronized alphameric field +
Commercial output editing a non-synchronized numeric field
Scientific output editing a non-synchronized numeric field) x
(Proportion of non-synchronized fields).

As these values have already been produced during this computation (PR 0902), or are included in the Engineering Vector (EV 1409, EV 1410, EV 1411, EV 1415, EV 1416, EV 1417), the value of TEV 09 (G, IO, S) is:

1/3((EV 1409) + (EV 1410) + (EV 1411)) x (PR 0902) +

1/3((EV 1415) + (EV 1416) + (EV 1417)) x (1 - PR 0902)

= 1/3((_______) + (_______) + (_______)) x (_______) +

1/3((_______) + (_______) + (_______)) x (_______)

= __________ microseconds.

Pre-computation of synchronization of alphameric data under General/I-O Critical/ Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that synchronization will be inversely proportional to the word length in alphameric characters.

As this value is included in the Engineering Vector (EV 1203), the value of the element concerned is:

1/(EV 1203) = 1/(________) = ________.

Pre-computation of the time taken to EDIT A FLEXIBLE NUMERIC FIELD DURING OUTPUT under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,.
2. Synchronization will be complete,
3. Output fields are equally likely to require scientific or commercial editing,

and then computing the following: One-half of the time involved when running time is minimized in:

(Commercial output editing a synchronized numeric field + Scientific output editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1413, EV 1414), the value of TEV 10 (CP) is:

1/2((EV 1413) + (EV 1414))

= 1/2((_______) + (_______))

= ___________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE NUMERIC FIELD DURING OUTPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average numeric field is 5 digits,
2. Synchronization will be complete,
3. Output fields are equally likely to require scientific or commercial editing,

and then computing the following: One-half of the time involved when programming is minimized in:

(Commercial output editing a synchronized numeric field + Scientific output editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1410, EV 1411), the value of TEV 10 (G, IO, S) is:

1/2((EV 1410) + (EV 1411))

= 1/2((________) + (________))

= __________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE ALPHAMERIC FIELD DURING OUTPUT under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,
2. Synchronization will be complete,
3. Fields not presently known to be numeric are still equally likely to require numeric handling,
4. Output fields are equally likely to require alphabetic, scientific, or commercial editing,

and then computing the following: One-third of the time involved when running time is minimized in:

(Output editing a synchronized alphameric field +
Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1412, EV 1413, EV 1414), the value of TEV 11 (CP) is:

1/3((EV 1412) + (EV 1413) + (EV 1414))

= 1/3((________) + (________) + (________))

= __________ microseconds.

Pre-computation of the time taken to EDIT A FLEXIBLE ALPHAMERIC FIELD DURING OUTPUT under General/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that:

1. The average alphameric field is 11 characters,

2. Synchronization will be complete,

3. Fields not presently known to be numeric are still equally likely to require numeric handling,

4. Output fields are equally likely to require alphabetic, scientific, or commercial editing,

and then computing the following: One-third of the time involved when programming is minimized in:

(Output editing a synchronized alphameric field +
Commercial output editing a synchronized numeric field +
Scientific output editing a synchronized numeric field).

As these values are included in the Engineering Vector (EV 1409, EV 1410, EV 1411), the value of TEV 11 (G, IO, S) is:

1/3((EV 1409) + (EV 1410) + (EV 1411))

= 1/3((________) + (________) + (________))

= ____________ microseconds.

Pre-computation of time taken to CONTROL THE PROCESSING OF A RECORD under General/CP Critical/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that, for each record, one comparison, one addition step, and one index register step and test will be involved.

As these values are included in the Engineering Vector (EV 1301, EV 1311, EV 1305), the value of TEV 12 (G, CP, IO, S) is:

(EV 1301) + (EV 1311) + (EV 1305)

= (_______) + (_______) + (_______)

= ___________ microseconds.

Pre-computation of time taken per character to CONTROL THE MOVEMENT OF A RECORD under General/CP Critical/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that either the record will be moved by means of an internal transfer operation, or scatter read and gather write facilities will be used, depending upon which technique is the more efficient.

As these values are included in the Engineering Vector (EV 1310, EV 1612), the value of TEV 13 (G, CP, IO, S) is:

The lesser of (EV 1310) or (EV 1612);

i.e., the lesser of (______) or (______)

= __________ microseconds.

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER ALPHAMERIC CHARACTER under General/Space Critical conditions.

Pre-requisites: PR 1401.

Method

This element can be arrived at by assuming that PR 1401 correctly estimates the packing efficiency and then computing the following:

(CP load per character) + (packing efficiency).

As these values have already been produced during this computation (PR 1401) or are included in the Engineering Vector (EV 1606), the value of TEV 14 (G, S) is:

(EV 1606) ÷ (PR 1401)

= (_______) ÷ (_______)

= ___________ microseconds per character.
(evaluated to 3 significant figures)

Pre-computation of packing efficiency for alphameric characters when operating under General/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under these conditions is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1401	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is _______, the value of PR 1401 is ________.

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER ALPHAMERIC CHARACTER under CP Critical conditions.

Pre-requisites: PR 1402.

Method

This element can be arrived at by assuming that PR 1402 correctly estimates the packing efficiency and then computing the following:

(CP load per character) ÷ (packing efficiency).

As these values have already been produced during this computation (PR 1402) or are included in the Engineering Vector (EV 1606), the value of TEV 14 (CP) is:

(EV 1606) ÷ (PR 1402)

= (_______) ÷ (_______)

= __________microseconds per character.
(evaluated to 3 significant figures)

Pre-computation of packing efficiency for alphameric characters when operating under CP Critical conditions.

<u>Pre-requisites:</u> None.

<u>Method</u>

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1402	1.00	0.80	0.75	0.75	0.70	0.70	0.70	0.65	0.65	0.65	0.60	0.60

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is _______, the value of PR 1402 is _______

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER ALPHAMERIC CHARACTER under I-O Critical conditions.

Pre-requisites: PR 1403.

Method

This element can be arrived at by assuming that PR 1403 correctly estimates the packing efficiency and then computing the following:

(CP load per character) ÷ (packing efficiency).

As these values have already been produced during this computation (PR 1403) or are included in the Engineering Vector (EV 1606), the value of TEV 14 (IO) is:

(EV 1606) ÷ (PR 1403)

= (_______) ÷ (_______)

= ___________ microseconds per character.
(evaluated to 3 significant figures)

Pre-computation of packing efficiency for alphameric characters when operating under I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1403	1.00	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85	0.80	0.75

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is _______, the value of PR 1403 is________.

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER DECIMAL DIGIT under General/Space Criteria conditions.

Pre-requisites: PR 1501, PR 1502.

Method

This element can be arrived at by assuming that PR 1502 correctly estimates the packing efficiency and then computing the following:

(CP load per digit) $\div$ (packing efficiency).

As these values have already been produced during this computation (PR 1501, PR 1502), the value of TEV 15 (G, S) is:

(PR 1501) $\div$ (PR 1502)

= (______) $\div$ (______)

= ____________________ microseconds per digit.
(evaluated to 3 significant figures)

Pre-computation of central processor time used per decimal digit read or written under all conditions.

Pre-requisites: None.

Method

This element can be arrived at by dividing the central processor time used per alphameric character read or written by the number of decimal digits per alphameric character in internal representation.

As these values are included in the Engineering Vector (EV 1606, EV 1607), the value of PR 1501 is:

(EV 1606) ÷ (EV 1607)

= (______) ÷ (______)

= ____________ microseconds per digit.

Pre-computation of packing efficiency for decimal digits when operating under General/ Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under these conditions is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	$\geq$ 12
Value of PR 1502	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value of PR 1502 is ________.

Pre-computation of magnetic tape load on central processor per decimal digit under CP Critical conditions.

Pre-requisites: PR 1501, PR 1503.

Method

This element can be arrived at by assuming that PR 1503 correctly estimates the packing efficiency and then computing the following:

(CP load per digit) $\div$ (packing efficiency).

As these values have already been produced during this computation (PR 1501, PR 1503), the value of TEV 15 (CP) is:

(PR 1501) $\div$ (PR 1503)

= (_______) $\div$ (_______)

= _______________ microseconds per digit.
(evaluated to 3 significant figures)

Pre-computation of packing efficiency for decimal digits when operating under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1503	1.00	0.80	0.75	0.75	0.70	0.70	0.70	0.65	0.65	0.65	0.60	0.60

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is ________, the value of PR 1503 is ________.

Pre-computation of magnetic tape load on central processor per decimal digit under I-O Critical conditions.

<u>Pre-requisites:</u> PR 1501, PR 1504.

<u>Method</u>

This element can be arrived at by assuming that PR 1504 correctly estimates the packing efficiency and then computing the following:

(CP load per digit) $\div$ (packing efficiency).

As these values have already been produced during this computation (PR 1501, PR 1504), the value of TEV 15 (IO) is:

(PR 1501) $\div$ (PR 1504)

= (______) $\div$ (______)

= ________________ microseconds per digit.
(evaluated to 3 significant figures)

Pre-computation of packing efficiency for decimal digits when operating under I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1504	1.00	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85	0.80	0.75

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value of PR 1504 is ________.

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER CARD IMAGE under General/CP Critical/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by multiplying the load on the central processor per alphameric character by 80.

As this value is included in the Engineering Vector (EV 1606), the value of TEV 16 (G, CP, IO, S) is:

(EV 1606) x 80

= (_______) x 80

= _____________ microseconds per card image.
(evaluated to 3 significant figures)

Pre-computation of magnetic tape LOAD ON CENTRAL PROCESSOR PER LINE IMAGE under General/CP Critical/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by multiplying the load on the central processor per alphameric character by 120.

As this value is included in the Engineering Vector (EV 1606), the value of TEV 17 (G, CP, IO, S) is:

(EV 1606) x 120

= (   ) x 120

= ____________ microseconds per line image.
(evaluated to 3 significant figures)

Pre-computation of MAGNETIC TAPE PERFORMANCE ON ALPHAMERIC DATA under General conditions.

<u>Pre-requisites:</u> PR 1802, PR 1803.

<u>Method</u>

This element can be arrived at by assuming that PR 1802 correctly estimates the block size to be used and PR 1803 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 1802, PR 1803) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) x (PR 1803)) x ((PR 1802) ÷ ((PR 1802) + (EV 1602)))

= ((________) x (________)) x ((________) ÷ ((________) + (________)))

= ____________________ characters per second.
(evaluated to 3 significant figures)

This gives a rate in characters/second. To transform this into the required microseconds/character form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ __________ = __________, which is TEV 18 (G).

Pre-computation of target block size for alphameric data when operating under General conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 2,000 characters, and will be used if it does not exceed 5% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If 2,000 $\leq$ (0.05 x (EV 1201)), then PR 1801 = 2,000.
Otherwise, PR 1801 = (0.05 x (EV 1201)).

Since EV 1201 = ______________,

(0.05 x (EV 1201)) = ___________.

Therefore, PR 1801 = ___________.

Pre-computation of working block size for alphameric data when operating under General conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1801.

Method

This element can be arrived at by assuming that the target block size, PR 1801, will be used where possible.

The quantities required are:

Minimum block size (EV 1603)
Maximum block size (EV 1605).

If (EV 1603) $\leq$ (PR 1801) and (EV 1605) $\geq$ (PR 1801), then
PR 1802 = PR 1801.

Otherwise, if (EV 1603) > (PR 1801), then PR 1802 = EV 1603;
or if (EV 1605) < (PR 1801), then PR 1802 = EV 1605.

Since (EV 1603) = ________________,

(EV 1605) = ________________,

and (PR 1801) = ________________,

the value of PR 1802 = ________________.

Pre-computation of packing efficiency for alphameric characters when operating under General conditions.

Pre-requisites: Nonc.

Method

This element can be arrivcd at by assuming that thc packing efficiency under this condition is relatcd solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1803	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is ______, the value of PR 1803 is ______.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON ALPHAMERIC DATA under CP Critical conditions.

Pre-requisites: PR 1805, PR 1806.

Method

This element can be arrived at by assuming that PR 1805 correctly estimates the block size to be used and PR 1806 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 1805, PR 1806) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) x (PR 1806)) x ((PR 1805) ÷ ((PR 1805) + (EV 1602)))

((________) x (________)) x ((________) ÷ ((________) + (________)))

= ________________characters per second.
(evaluated to 3 significant figures)

This gives a rate in characters/second. To transform this into the required microseconds/character form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ______________ = ______________, which is TEV 18 (CP).

Pre-computation of target block size for alphameric data when operating under CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 2,000 characters, and will be used if it does not exceed 5% of the memory size.

The quantity required is
memory size in alphameric characters (EV 1201).

If 2,000 $\leq$ (0.05 x (EV 1201)), then PR 1804 = 2,000.
Otherwise, PR 1804 = (0.5 x (EV 1201)).

Since EV 1201 = ________________,

(0.05 x (EV 1201)) = ________________.

Therefore, PR 1804 = ________________.

Pre-computation of working block size for alphameric data when operating under CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1804.

Method

This element can be arrived at by assuming that the target block size, PR 1804, will be used where possible.

The quantities required are:

Minimum block size (EV 1603)
Maximum block size (EV 1605).

If (EV 1603) $\leq$ (PR 1804) and (EV 1605) $\geq$ (PR 1804), then
PR 1805 = PR 1804.

Otherwise, if (EV 1603) > (PR 1804), then PR 1805 = EV 1603.

Since (EV 1603) = ________________,

(EV 1605) = ________________,

and (PR 1804) = ________________.

the value of PR 1805 = ________________.

Pre-computation of packing efficiency for alphameric characters when operating under CP Critical conditions.

Pre-requisites: None

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1806	1.00	0.80	0.75	0.70	0.70	0.70	0.70	0.70	0.65	0.65	0.60	0.60

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is ________, the value of PR 1806 is ________.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON ALPHAMERIC DATA under I-O Critical conditions.

Pre-requisites: PR 1808, PR 1809.

Method

This element can be arrived at by assuming that PR 1808 correctly estimates the block size to be used and PR 1809 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 1808, PR 1809) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) x (PR 1809)) x ((PR 1808) ÷ ((PR 1808) + (EV 1602)))

= ((_______) x (_______)) x ((_______) ÷ ((_______) + (_______)))

= ______________ characters per second.
(evaluated to 3 significant figures)

This gives a rate in characters/second. To transform this into the required microseconds/character form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 18 (IO).

Pre-computation of target block size for alphameric data when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 4,000 characters, and will be used if it does not exceed 15% of the memory size.

The quantity required is
memory size in alphameric characters (EV 1201).

If 4,000 $\leq$ (0.15 x (EV 1201)), then PR 1807 = 4,000.
Otherwise, PR 1807 = (0.15 x (EV 1201)).

Since EV 1201 = ________________,

(0.15) x (EV 1201)) = ________________.

Therefore, PR 1807 = ________________.

Pre-computation of working block size for alphameric data when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1807.

Method

This element can be arrived at by assuming that the target block size, PR 1807, will be used where possible.

The quantities required are:

Minimum block size (EV 1603)
Maximum block size (EV 1605).

If (EV 1603) $\leq$ (PR 1807) and (EV 1605) $\geq$ (PR 1807), then PR 1808 = PR 1807.

Otherwise, if (EV 1603) > (PR 1807), then PR 1808 = EV 1603; or if (EV 1605) < (PR 1807), then PR 1808 = EV 1605.

Since (EV 1603) = ________________,

(EV 1605) = ________________,

and (PR 1807) = ________________,

the value of PR 1808 = ________________.

Pre-computation of packing efficiency for alphameric characters when operating under I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1809	1.00	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85	0.80	0.75

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is _______, the value of PR 1809 is _______.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON ALPHAMERIC DATA under Space Critical conditions.

Pre-requisites: PR 1811, PR 1812.

Method

This element can be arrived at by assuming that PR 1811 correctly estimates the block size to be used and PR 1812 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 1811, PR 1812) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) x (PR 1812)) x ((PR 1811) ÷ ((PR 1811) + (EV 1602)))

= ((_______) x (_______)) x ((_______) ÷ ((_______) + (_______)))

= _______________ characters per second.
(evaluated to 3 significant figures)

This gives a rate in characters/second. To transform this into the required microseconds/character form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 18 (S).

Pre-computation of target block size for alphameric data when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 1,000 characters, and will be used if it does not exceed 5% of the memory size.

The quantity required is
memory size in alphameric characters (EV 1201).

If 1,000 $\leq$ (0.05 x (EV 1201)), then PR 1810 = 1,000.
Otherwise, PR 1810 = (0.05 x (EV 1201)).

Since EV 1201 = __________,

(0.05 x EV 1201)) = __________.

Therefore, PR 1810 = __________.

Pre-computation of working block size for alphameric data when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1810.

Method

This element can be arrived at by assuming that the target block size, PR 1810, will be used where possible.

The quantities required are:

Minimum block size (EV 1603)
Maximum block size (EV 1605).

If (EV 1603) $\leq$ (PR 1810) and (EV 1605) $\geq$ (PR 1810), then PR 1811 = PR 1810.

Otherwise, if (EV 1603) > (PR 1810), then PR 1811 = EV 1603; or if (EV 1605) < (PR 1810), then PR 1811 = EV 1605.

Since (EV 1603) = ______________,

(EV 1605) = ______________,

and (PR 1810) = ______________,

the value of PR 1811 = ______________.

Pre-computation of packing efficiency for alphameric characters under Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in alphameric characters and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1401	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in alphameric characters is included in the Engineering Vector (EV 1203) and is _______, the value of PR 1812 is ________.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON DECIMAL DATA under General conditions.

Pre-requisites: PR 1901, PR 1902, PR 1912, PR 1913.

Method

This element can be arrived at by assuming that PR 1912 correctly estimates the block size to be used and PR 1913 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size + (block size + gap cost)).

As these values have already been produced during this computation, the value of the element concerned is:

((PR 1901) x (PR 1913)) x ((PR 1912) ÷ ((PR 1912) + (PR 1902)))

= ((_______) x (_______)) x ((_______) ÷ ((_______) + (_______)))

________________digits per second.
(evaluated to 3 significant figures)

This gives a rate in digits/second. To transform this into the required microseconds/digit form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ __________ = ____________, which is TEV 19 (G).

Pre-computation of magnetic tape peak speed in decimal digits per second under all conditions.

Pre-requisites: None.

Method

This element can be arrived at by multiplying the peak speed in alphameric characters per second by the number of decimal digits per alphameric character on magnetic tape.

As these values are included in the Engineering Vector (EV 1601, EV 1608), the value of PR 1901 is:

(EV 1601) x (EV 1608)

= (_______) x (_______)

= _____________ digits per second.

Pre-computation of minimum gap cost in decimal digits under all conditions.

<u>Pre-requisites</u>: None.

<u>Method</u>

This element can be arrived at by multiplying the minimum gap cost in alphameric characters by the number of decimal digits per alphameric character on magnetic tape.

As these values are included in the Engineering Vector (EV 1602, EV 1608), the value of PR 1902 is:

(EV 1602) x (EV 1608)

= (_______) x (_______)

= ____________ digits.

Pre-computation of minimum block size in decimal digits under all conditions.

Pre-requisites: None.

Method

This element can be arrived at by multiplying the minimum block size in alphameric characters by the number of decimal digits per alphameric character on magnetic tape.

As these values are included in the Engineering Vector (EV 1603, EV 1608), the value of PR 1903 is:

(EV 1603) x (EV 1608)

= (_______) x (_______)

= _____________ digits.

Pre-computation of maximum block size in decimal digits under all conditions.

Pre-requisites: None.

Method

This element can be arrived at by multiplying the maximum block size in alphameric characters by the number of decimal digits per alphameric character on magnetic tape.

As these values are included in the Engineering Vector (EV 1605, EV 1608), the value of PR 1904 is:

(EV 1605) x (EV 1608)

= (_______) x (_______)

= ______________ digits.

Pre-computation of target block size for decimal data when operating under General conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 2,000 digits and will be used if it does not exceed 5% of the memory size.

The quantity required is memory size in decimal digits (EV 1202).

If 2,000 $\leq$ (0.05 x (EV 1202)), then PR 1911 = 2,000.
Otherwise, PR 1911 = (0.05 x (EV 1202)).

Since EV 1202 = ______________,

(0.05 x (EV 1202)) = ______________.

Therefore, PR 1911 = ______________.

Pre-computation of working block size for decimal data when operating under General conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1911, PR 1903, PR 1904

Method

This element can be arrived at by assuming that the target block size, PR 1911 will be used where possible.

The quantities required are:

Minimum block size (PR 1903)
Maximum block size (PR 1904).

If (PR 1903) $\leq$ (PR 1911) and (PR 1904) $\geq$ (PR 1911), then PR 1912 = PR 1911.

Otherwise, if (PR 1903) > (PR 1911), then PR 1912 = PR 1903; or if (PR 1904) < (PR 1911), then PR 1912 = PR 1904.

Since (PR 1903) = ______________,

(PR 1904) = ______________,

and (PR 1911) = ______________,

the value of PR 1912 = ______________.

Pre-computation of packing efficiency for decimal digits when operating under General conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1913	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value fo PR 1913 is ________.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON DECIMAL DATA under CP Critical conditions.

Pre-requisites: PR 1901, PR 1902, PR 1915, PR 1916.

Method

This element can be arrived at by assuming that PR 1915 correctly estimates the block size to be used and PR 1916 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation, the value of the element concerned is:

((PR 1901) x (PR 1916)) x ((PR 1915) ÷ ((PR 1915) + (PR 1902)))

= ((________) x (________)) x ((________) ÷ ((________) + (________)))

= ________________ digits per second.
(evaluated to 3 significant figures)

This gives a rate in digits/second. To transform this into the required microseconds/digits form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 19 (CP).

Pre-computation of target block size for decimal data when operating under CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 2,000 digits, and will be used if it does not exceed 5% of the memory size.

The quantity required is memory size in decimal digits (EV 1202).

If 2,000 $\leq$ (0.05 x (EV 1202)), then PR 1914 = 2,000.
Otherwise, PR 1914 = (0.05 x (EV 1202)).

Since EV 1202 = ______________,

(0.05 x (EV 1202)) = ______________.

Therefore, PR 1914 = ______________.

Pre-computation of working block size for decimal data when operating under CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1914, PR 1903, PR 1904.

Method

This element can be arrived at by assuming that the target block size, PR 1914, will be used where possible.

The quantities required are:

Minimum block size (PR 1903)
Maximum block size (PR 1904).

If (PR 1903) $\leq$ (PR 1914) and (PR 1904) $\geq$ (1914), then PR 1915 = PR 1914.

Otherwise, if (PR 1903) > (PR 1914), then PR 1915 = PR 1903; or if (PR 1904) < (PR 1914), then PR 1915 = PR 1904.

Since (PR 1903) = ______________,

(PR 1904) = ______________,

and (PR 1914) = ______________,

the value of PR 1915 = ______________.

Pre-computation of packing efficiency for decimal digits when operating under CP Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1916	1.00	0.80	0.75	0.75	0.70	0.70	0.70	0.65	0.65	0.65	0.60	0.60

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value of PR 1916 is ________.

Pre-computation of target block size for decimal data when operating under I-O Critical conditions, for use in determining magnetic tape performance.

<u>Pre-requisites:</u> None.

<u>Method</u>

This element can be arrived at by assuming that the desirable block size is 4,000 characters, and will be used if it does not exceed 15% of the memory size.

The quantity required is
memory size in decimal digits (EV 1202).

If 4,000 $\leq$ (0.05 x (EV 1202)), then PR 1917 = 4,000.
Otherwise, PR 1917 = (0.15 x (EV 1202)).

Since EV 1202 = ____________,

(0.15 x (EV 1202)) = ____________.

Therefore, PR 1917 = ____________.

Pre-computation of working block size for decimal data when operating under I-O Critical conditions, for use in determining magnetic tape performance.

<u>Pre-requisites:</u> PR 1917, PR 1903, PR 1904.

<u>Method</u>

This element can be arrived at by assuming that the target block size, PR 1917, will be used where possible.

The quantities required are:

Minimum block size (PR 1903)
Maximum block size (PR 1904).

If (PR 1903) $\leq$ (PR 1917) and (PR 1904) $\geq$ (PR 1917), then PR 1918 = PR 1917.

Otherwise, if (PR 1903) > (PR 1917), then PR 1918 = PR 1903; or if (PR 1904) < (PR 1917), then PR 1918 = PR 1904.

Since (PR 1903) = ________________,

(PR 1904) = ________________,

and (PR 1917) = ________________,

the value of PR 1918 = ________________.

Pre-computation of packing efficiency for decimal digits when operating under I-O Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	≥ 12
Value of PR 1913	1.00	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85	0.80	0.75

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value of the PR 1919 is ________.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON DECIMAL DATA under Space Critical conditions.

Pre-requisites: PR 1901, PR 1902, PR 1921, PR 1922.

Method

This element can be arrived at by assuming that PR 1921 correctly estimates the block size to be used and PR 1922 the packing efficiency, and then computing the following:

(peak speed x packing efficiency) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation, the value of the element concerned is:

((PR 1901) x (PR 1922)) x ((PR 1921) ÷ ((PR 1921) + (PR 1902)))

= ((_______) x (_______)) x ((_______) ÷ ((_______) + (_______)))

= ____________ digits per second.
(evaluated to 3 significant figures)

This gives a rate in digits/second. To transform this into the required microseconds/digit form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 19 (S).

Pre-computation of target block size for decimal data when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 1,000 characters, and will be used if it does not exceed 5% of the memory size.

The quantity required is
memory size in decimal digits (EV 1202).

If 1,000 $\leq$ (0.05 x (EV 1202)), then PR 1920 = 1,000.
Otherwise, PR 1920 = (0.05 x (EV 1202)).

Since EV 1202 = ________________,

(0.05 x (EV 1202)) = ________________.

Therefore, PR 1920 = ________________.

Pre-computation of working block size for decimal data when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 1921, PR 1903, PR 1904.

Method

This element can be arrived at by assuming that the target block size, PR 1920, will be used where possible.

The quantities required are:

Minimum block size (PR 1903)
Maximum block size (PR 1904).

If (PR 1903) $\leq$ (PR 1920) and (PR 1904) $\geq$ (PR 1920), then PR 1921 = PR 1920.

Otherwise, if PR 1903) > (PR 1920), then PR 1921 = PR 1903; or if (PR 1904) < (PR 1920), then PR 1921 = PR 1904.

Since (PR 1903) = ________________,

(PR 1904) = ________________,

and (PR 1920) = ________________,

the value of PR 1921 = ________________.

Pre-computation of packing efficiency for decimal digits when operating under Space Critical conditions.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the packing efficiency under this condition is related solely to the word size in decimal digits and is correctly given in the table below.

Word Length	1	2	3	4	5	6	7	8	9	10	11	$\geq$ 12
Value of PR 1922	1.00	0.85	0.80	0.80	0.75	0.75	0.75	0.75	0.70	0.70	0.65	0.60

As the word length in decimal digits is included in the Engineering Vector (EV 1204) and is _______, the value of PR 1922 is ________.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON CARD IMAGES under General/CP Critical conditions.

Pre-requisites: PR 2002.

Method

This element can be arrived at by assuming that PR 2002 correctly estimates the block size to be used, and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2002) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 80) x (80 x (PR 2002)) ÷ (80 x (PR 2002) + (EV 1602))

= ((_______) ÷ 80) x (80 x (_______)) ÷ (80 x (_______) + (_______))

= __________ card images per second.
(evaluated to 3 significant figures)

This gives a rate in card images/second. To transform this into the required microseconds/card image form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ _______ = _______, which is TEV 20 (G, CP).

Pre-computation of target block size for card images when operating under General/ CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 25 cards, and will be used if it does not exceed 5% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If $2{,}000 \leq (0.05 \times (\text{EV } 1201))$, then PR 2001 = 25.
Otherwise, PR 2001 = $(0.05 \times (\text{EV } 1201)) \div 80$, to the nearest integer.

Since EV 1201 = _______________,

(0.05 x (EV 1201)) = _______________.

Therefore, PR 2001 = _______________ cards.

Pre-computation of working block size for card images when operating under General/ CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2001.

Method

This element can be arrived at by assuming that the target block size, PR 2001, will be used wherever it does not exceed the maximum blocking factor for card image input using standard routines, EV 1610.

If (EV 1610) $\geq$ (PR 2001), then PR 2002 = PR 2001.
If (EV 1610) $<$ (PR 2001), then PR 2002 = EV 1610.

Since EV 1610 = __________,

and PR 2001 = __________,

the value of PR 2002 = ____________ cards.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON CARD IMAGES under I-O Critical conditions.

<u>Pre-requisites</u>: PR 2004.

<u>Method</u>

This element can be arrived at by assuming that PR 2004 correctly estimates the block size to be used, and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2004) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 80) x (80 x (PR 2004)) ÷ (80 x (PR 2004) + (EV 1602))

((_______) ÷ 80) x (80 x (_______)) ÷ (80 x (_______) + (_______))

= ________________ card images per second.
(evaluated to 3 significant figures)

This gives a rate in card images/second. To transform this into the required micro-seconds/card image form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ __________ = __________, which is TEV 20 (IO).

Pre-computation of target block size for card images when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 50 cards, and will be used if it does not exceed 15% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If $4,000 \leq (0.15 \times (\text{EV } 1201))$, then PR 2003 = 50.
Otherwise, PR 2003 = $(0.15 \times (\text{EV } 1201)) \div 80$, to the nearest integer.

Since EV 1201 = ______________,

$(0.15 \times (\text{EV } 1201))$ = ______________.

Therefore, PR 2003 = ______________ cards.

Pre-computation of working block size for card images when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2003.

Method

This element can be arrived at by assuming that the target block size, PR 2003, will be used wherever it does not exceed the maximum blocking factor for card image input using standard routines, EV 1610.

If (EV 1610) $\geq$ (PR 2003), then PR 2004 = PR 2003.
If (EV 1610) $<$ (PR 2003), then PR 2004 = EV 1610.

Since EV 1610 = ______________,

and PR 2003 = ______________,

the value of PR 2004 = ______________ cards.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON CARD IMAGES under Space Critical conditions.

Pre-requisites: PR 2006.

Method

This element can be arrived at by assuming that PR 2006 correctly estimates the block size to be used, and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2006) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 80) x (80 x (PR 2006)) ÷ (80 x (PR 2006) + (EV 1602))

= ((_______) ÷ 80) x (80 x (_______)) ÷ (80 x (_______) + (_______))

= _______________ card images per second.
(evaluated to 3 significant figures)

This gives a rate in card images/second. To transform this into the required micro-seconds/card image form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ __________ = __________, which is TEV 20 (S).

Pre-computation of target block size for card images when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 12 cards, and will be used if it does not exceed 2% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If $960 \leq (0.02 \times (\text{EV } 1201))$, then PR 2005 = 12.
Otherwise, PR 2005 = $(0.02 \times (\text{EV } 1201)) \div 80$, to the nearest integer.

Since EV 1201 = ______________,

(0.02 x (EV 1201)) = ______________.

Therefore, PR 2005 = ______________ cards.

Pre-computation of working block size for card images when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2005.

Method

This element can be arrived at by assuming that the target block size, PR 2005, will be used wherever it does not exceed the maximum blocking factor for card image input using standard routines, EV 1610.

If (EV 1610) $\geq$ (PR 2005), then PR 2006 = PR 2005.
If (EV 1610) $<$ (PR 2005), then PR 2006 = EV 1610.

Since EV 1610 = ______________,

and PR 2005 = ______________,

the value of PR 2006 = ______________ cards.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON LINE IMAGES under General CP Critical conditions.

Pre-requisites: PR 2102.

Method

This element can be arrived at by assuming that PR 2102 correctly estimates the block size to be used, and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2102) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 120) x (120 x (PR 2102)) ÷ (120 x (PR 2102) + (EV 1602))

= ((_______) ÷ 120) x (120 x (_______)) ÷ (120 x (_______) + (_______))

= ________________ line images per second.
(evaluated to 3 significant figures)

This gives a rate in line images/second. To transform this into the required microseconds/line image form, divide it into 1, 000, 000, again to 3 significant figures; i.e., 1, 000, 000 ÷ __________ = __________, which is TEV 21 (G, CP).

Pre-computation of target block size for line images when operating under General/ CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 16 line images, and will be used if it does not exceed 5% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If 1, 920 $\leq$ (0. 05 x (EV 1201)), then PR 2101 = 16.
Otherwise, PR 2101 = (0. 05 x (EV 1201)) $\div$ 120, to the nearest integer.

Since EV 1201 = ______________,

(0. 05 x (EV 1201)) = ______________,

Therefore, PR 2101 = ______________ lines.

Pre-computation of working block size for line images when operating under General/ CP Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2101

Method

This element can be arrived at by assuming that the target block size, PR 2101, will be used wherever it does not exceed the maximum blocking factor for line image output using standard routines, EV 1611.

If (EV 1611) $\geq$ (PR 2101), then PR 2102 = PR 2101.
If (EV 1611) < (PR 2101), then PR 2102 = EV 1611.

Since EV 1611 = ______________,

and PR 2101 = ______________,

the value of PR 2102 = ______________ lines.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON LINE IMAGES under I-O Critical conditions.

Pre-requisites: PR 2104.

Method

This element can be arrived at by assuming that PR 2104 correctly estimates the block size to be used and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2104) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 120) x (120 x (PR 2104)) + (120 x (PR 2104) + (EV 1602))

= ((________) ÷ 120) x (120 x (________)) ÷ (120 x (________) + (________))

= ________________ line images per second.
(evaluated to 3 significant figures)

This gives a rate in line images/second. To transform this into the required microseconds/line image form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ______________ = ______________, which is TEV 21 (IO).

Pre-computation of target block size for line images when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: None.

Method

This element can be arrived at by assuming that the desirable block size is 33 line images, and will be used if it does not exceed 15% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If $3,960 \leq (0.15 \times (\text{EV } 1201))$, then PR 2103 = 33.
Otherwise, PR 2103 = $(0.15 \times (\text{EV } 1201)) \div 120$, to the nearest integer.

Since EV 1201 = ________________,

(0.15 x (EV 1201)) = ________________.

Therefore, PR 2103 = ________________lines.

Pre-computation of working block size for line images when operating under I-O Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2104.

Method

This element can be arrived at by assuming that the target block size, PR 2103, will be used wherever it does not exceed the maximum blocking factor for line image output using standard routines, EV 1611.

If (EV 1611) $\geq$ (PR 2103), then PR 2104 = PR 2103.
If (EV 1611) $<$ (PR 2103), then PR 2104 = EV 1611.

Since EV 1611 = ____________________,

and PR 2103 = ____________________,

the value of PR 2104 = ____________________ lines.

Pre-computation of MAGNETIC TAPE PERFORMANCE ON LINE IMAGES under Space Critical conditions.

Pre-requisites: PR 2106

Method

This element can be arrived at by assuming that PR 2106 correctly estimates the block size to be used, and then computing the following:

(peak speed) x (block size ÷ (block size + gap cost)).

As these values have already been produced during this computation (PR 2106) or are included in the Engineering Vector (EV 1601, EV 1602), the value of the element concerned is:

((EV 1601) ÷ 120) x (120 x (PR 2106)) ÷ (120 x (PR 2106) + (EV 1602))

((________) ÷ 120) x (120 x (________)) ÷ (120 x (________) + (________))

________________ line images per second.
(evaluated to 3 significant figures)

This gives a rate in line images/second. To transform this into the required microseconds/line image form, divide it into 1,000,000, again to 3 significant figures; i.e., 1,000,000 ÷ ____________ = ____________, which is TEV 21 (S).

Pre-computation of target block size for line images when operating under Space Critical conditions, for use in determining magnetic tape performance.

<u>Pre-requisites</u>: None.

<u>Method</u>

This element can be arrived at by assuming that the desirable block size is 8 line images and will be used if it does not exceed 2% of the memory size.

The quantity required is memory size in alphameric characters (EV 1201).

If $960 \leq (0.02 \times (\text{EV } 1201))$, then PR 2105 = 8.
Otherwise, PR 2105 = $(0.02 \times (\text{EV } 1201)) + 120$, to the nearest integer.

Since EV 1201 = ______________,

(0.02 x (EV 1201)) = ______________.

Therefore, PR 2105 = ______________ lines.

Pre-computation of working block size for line images when operating under Space Critical conditions, for use in determining magnetic tape performance.

Pre-requisites: PR 2105.

Method

This element can be arrived at by assuming that the target block size, PR 2105, will be used wherever it does not exceed the maximum blocking factor for line image output using standard routines, EV 1611.

If (EV 1611) $\geq$ (PR 2105), then PR 2106 = PR 2105.
If (EV 1611) < (PR 2105), then PR 2106 = EV 1611.

Since EV 1611 = ______________,

and PR 2105 = ______________.

the value of PR 2106 = ______________ lines.

Pre-computation of SIMULTANEITY RULE NUMBER under General/CP Critical/I-O Critieal/Space Critical conditions.

Pre-requisites: None

Method

This element can be arrived at by assuming the following rules:

Rule 1 requires that reading and writing ehannels are separate, and that processing oeeupies one channel.
Rule 2 requires that reading and writing channels are separate, and that proeessing is independent.
Rule 3 requires that reading and writing ehannels are together, and that processing oceupies one channel.
Rule 4 requires that reading and writing ehannels are together, and that proeessing is independent.

As the required values are ineluded in the Engineering Veetor (EV 501 through EV 506), the value of TEV 22 (G, CP, IO, S) is:

1 if (EV 1501) = (EV 1503) = 1/2(EV 1505) = ((EV 1502) - 1)) = (EV 1504 -1) = 1/2((EV 1506) -1).

2 if (EV 1501) = (EV 1502) = (EV 1503) = (EV 1504) = 1/2(EV 1505) = 1/2(EV 1506).

3 if (EV 1501) = (EV 1503) = (EV 1505) = ((EV 1502) - 1) = ((EV 1504) - 1) = ((EV 1506) - 1).

4 if (EV 1501) = (EV 1502) = (EV 1503) = (EV 1504) = (EV 1505) = (EV 1506).

Sinee (EV 1501) = ________________,
(EV 1502) = ________________,
(EV 1503) = ________________,
(EV 1504) = ________________,
(EV 1505) = ________________,
and (EV 1506) = ________________;

TEV 22 (G, CP, IO, S) = ________________.

Pre-computation of SIMULTANEITY PARAMETER NUMBER when operating under General/CP Critical/I-O Critical/Space Critical conditions.

Pre-requisites: None.

Method

This element specifies the maximum number of simultaneous magnetic tape read operations that can be executed while internal processing is proceeding.

This value is included in the Engineering Vector (EV 1501). Therefore:

TEV 23(G, CP, IO, S) = (EV 1501) = ____________.

TRANSFORMED ENGINEERING VECTOR

TRANSFORMED ENGINEERING VECTOR

TRANSFORMED ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE UNDER GENERAL AND SPECIFIC LIMITING FACTORS			
		GEN	CP	I/O	SPACE
01	Edit a fixed numeric field during input				
02	Edit a fixed alphameric field during input				
03	Edit a flexible numeric field during input				
04	Edit a flexible alphameric field during input				
05	Simple Update Operation				
06	Complex Update				
07	Table Reference Time				
08	Edit a fixed numeric field during output				
09	Edit a fixed alphameric field during output				
10	Edit a flexible numeric field during output				
11	Edit a flexible alphameric field during output				
12	Control the processing of a record				
13	Control the movement of a record				
14	Load on central processor per alphameric character				
15	Load on central processor per decimal digit				
16	Load on central processor per card image				
17	Load on central processor per line image				
18	Magnetic tape performance on alphameric data				
19	Magnetic tape performance on decimal data				
20	Magnetic tape performance on card images				
21	Magnetic tape performance on line images				
22	Simultaneity rule number				
23	Simultaneity parameter number				

APPENDIX II

GUIDE AND FORMS FOR COMPLETION OF TRANSFORMED FUNCTIONAL VECTOR OF THE "VECTOR" ESTIMATING PROCESS

1. INTRODUCTION

The VECTOR method uses independent sets of numbers (Vectors) to represent the important characteristics of each computer and each application. The Vectors are used both to establish and time the optimum method on a specific computer system. The Vectors themselves are normally self-sufficient and no other knowledge of either the problem or the computer is assumed.

In these circumstances it can be seen that the creation of these Vectors is an unusually responsible process. This document is concerned with the preparation of the Functional Vector, shows the procedural steps needed, and attempts to illustrate the requirements and the opportunities given to allow each computer system to be shown at its honest best.

2. BACKGROUND AND PREPARATION INSTRUCTIONS — FUNCTIONAL SPECIFICATIONS

In preparing the specifications of the problem to be processed through the computer, we have constantly kept in mind the problem facing the analyst who will have to gather and supply the information. Our aim has been to specify data specifications in such form as to enable an evaluation to be performed even if precise knowledge of the application processes is not available for each process segment.

Many things enter into a computer systems evaluation, such as the budget available; the expected workload of the selected key job (or jobs); the necessary work which must be done in other areas to complete the application; how much unloaded computer time will be available when the overall application is completed for work that can be profitably placed on the computer system, and if so, what is the value of this work. All these are questions which are pertinent to a computer evaluation, since these figures will establish a basis for weighting the expected running times against the total system operating time.

None of these questions directly affect the actual details of the application itself, and therefore, in this specification, they are considered separately. The queries are included on the "Functional Specification — Application Background" sheet, one of which should be completed for each key job specification. A certain amount of explanation is given with each sheet, to enable easy understanding of the data required, and the reason for its need.

3. BACKGROUND AND PREPARATION INSTRUCTIONS FOR THE TRANSACTION FILE

3.1 General

The Transaction File contains all the new information which is to be used to update the Master File, create reports, etc. In general, the information needed relative to the transaction data to approximate a computer performance is related to three factors:

(1) The sequence in which the transactions are available.

(2) The description of each type of transaction.

(3) The work involved in processing each type of transaction.

The specification for (1) can be separated from the others, and is included on a separate sheet. The details of each of the specification queries is intended to be self-explanatory. The following pages contain detailed examples of how the individual items are to be completed — always bearing in mind that one sheet is to be completed for each transaction record type. On each sheet, details are first sought regarding the physical appearance of the file, how many, of what size records, etc.

Next, more complicated specifications are requested. These deal with the estimates of selected programming steps required to process each transaction type. The work-load is divided into three types of work, arbitrarily called; "Simple Update," "Complex Update," and "Table Reference." These have been chosen because it is not possible to estimate the number of multiplications a process will require per transaction from only a knowledge of the number of fields to be involved in the processing needed. Equally, the computers Add/Multiply or Add/Table Reference ratios vary greatly so that distinction must be made between these types of operations. In this they differ from operations involving subtraction, comparisons, divisions, etc., which can be regarded as additions or multiplications respectively without imperiling the accuracy of the results. The choice of these three processing categories represents a compromise on the minimum usable data and normal methods of analysis leading to inaccuracies. We do not feel that a systems analyst has a detailed knowledge of the number of arithmetic or logical operations to be performed. Therefore, the provision of categories of operation types is also intended to make it easier to indicate relative degrees of varying operation-type levels.

Due to the absence of detailed programming specifications of any problem, some degree of error in calculation of estimated running time is permissible. Our main problem has been to avoid the introduction of typical or average numbers for cases where individual problem and machine conditions demand actions that are significantly different from those governing general operating conditions. Examples of such actions might be the packing of magnetic tape formats through use of binary instead of decimal number representation; the loose packing (format arrangement) of magnetic tape records in order to eliminate editing within the central processor; the possible dynamic variation in record sizes of magnetic tape master file records; etc.

In considering the individual specifications which are required for each record type (remember that separate record types for all files are specified on separate pages), it is unlikely that all the records of the specific type go through the same process. Some will go through Process A, some through Process B as well. To allow for this there is space in FS 370-390 for four separate "processes" to be described. For instance, if in processing an inventory receipt transaction it is required to price the value of the merchandise in some cases, but not in others, such a situation might arise where some goods are received on consignment as opposed to those which are purchased. This would indicate that each day 10,000 issue vouchers were to be processed; 9,500 of them would involve only 2 addition, subtraction, or comparison operations, and 500 would involve 3 additions and a multiplication or division.

	PROCESS NAME:	Receipt	Price		
FS 350	% of records using this work path.	9,500	500		
FS 360	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)	2	3		
FS 370	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)	0	1		
FS 380	Average no. of decimal digits in the numeric operands used during the process.	5	5		
FS 390	No. of associated values (A) extracted from tables in execution of the work path processing.	A T	A T	A T	A T

It can be seen from this, that how the processes are split up is a matter of choice. Exactly the same results would occur if the specifications had been filled out like this:

	PROCESS NAME:	Receipt	Price		
FS 350	% of records using this work path.	10,000	500		
FS 360	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)	2	1		
FS 370	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)	0	1		

In each case, some 20,000 equivalent additions, and 500 equivalent multiplications would be timed when the process was being evaluated. It is preferred, but not always practical, that the same method be used; i. e., any logically common path is shown in one place only.

The table reference task involves the operation concerned with obtaining data from any size table. For instance, what time is the first train to Washington from New York after 6:00 a. m.; what is the place referred to by code 1A72; how many C589326ABC's are present in D675281PQR; what was the local currency exchange rate in Thailand last Monday, etc.

The data is requested in two parts:

(1) How many items are to be taken from tables?

(2) What size tables are they to be taken from?

and a special note is made that if the table has under 50 entries, the table size is not needed. For tables of under 50 items, it is not necessarily true that table searching or list processing techniques will always be used. In this case, the number of items will provide sufficient information for the Functional Vector.

An example might be where the location of a consignee has to be determined, and when this is determined, the method of shipping the goods must be established. Assuming that there is 4,000 possible consignees, and that there are six shipping methods, the previous example, completely filled out, might well look like this:

<table>
<tr><td></td><td>PROCESS NAME:</td><td colspan="2">Receipt</td><td colspan="2">Price</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>FS 350</td><td>% of records using this work path.</td><td colspan="2">10,000</td><td colspan="2">500</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>FS 360</td><td>No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>FS 370</td><td>No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>FS 380</td><td>Average no. of decimal digits in the numeric operands used during the process.</td><td colspan="2">5</td><td colspan="2">5</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td rowspan="5">FS 390</td><td rowspan="5">No. of associated values (A) extracted from tables in execution of the work path processing, arranged by table size involved (T). (Ignore T if tables have less than 50 entries.)</td><td>A</td><td>T</td><td>A</td><td>T</td><td>A</td><td>T</td><td>A</td><td>T</td></tr>
<tr><td>1</td><td>-</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>4000</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

This implies that each receipt transaction needs to have two arguments to be derived during its processing. Note that the comparisons which might well be needed in order to extract the information from the table are not shown separately. This work is implied in FS 390.

4. BACKGROUND AND PREPARATION OF INSTRUCTIONS FOR THE MASTER FILE

4.1 General

The Master File is described in a manner similar to the Transaction File: that is, one general sheet plus one Master File Record Sheet for each type of Master File Record. The Master File Record Sheet contains specifications of a few simple details, which are self-explanatory. The major details are included with the individual Master File Record Sheets.

In general, these Record Sheets reflect the composition of the file. Thus, the first three specifications include the number of records, the average number of characters per record, the average number of numeric digits per record. These should cause no problem provided that it is recalled that we are using one typical cycle as a test case. Thus while the number of records in the Master File certainly varies from day to day, it has an average value. This is the value to be entered in the specifications. Similarly, while there may be considerable variation in the size of individual records (relative to variable length record files), there should only be one average size on any specific day. In cases where variable record sizes are treated as headers and trailers, each trailer type should be treated as a separate type within the Master File.

After describing the makeup of the Master File Records, the next specifications relate to the work induced by each individual record. Very frequently, a Master File record will not induce any work whatsoever, and in these cases no entries need be made here. In such cases, all the processing involved will have been described either under the Transaction or Report files.

However, in some cases, work is induced by the Master File. Typically these could be:

(1) Where an inventory record was cyclically checked to provide for a re-order.

(2) Where a number of transactions were posted against a single Master File Record. In this case, any final checks made on the contents of the Master File record (again perhaps for re-order) might be made only once, at the end of the process instead of once per transaction.

(3) Where a summary report is being produced.

Where there is such work, it is described exactly as in the case of the Transaction File records, and reference should be made to that description for details as to methods.

Care should be taken to ensure that any one action is not described in both places, as this will lead to inaccuracies in the final estimates.

5. BACKGROUND AND PREPARATION INSTRUCTIONS FOR THE REPORT FILE

5.1 General

Again, the description and procedure are similar to those of the previous file: One General Description sheet, followed by a separate sheet for each type of report. The former simply checks the desired order of the Report File, its medium, the number of report formats, etc.

The Report Description sheet is made out for each Report type. In this it is not normally necessary to break down headings, subtotals, combinations, final totals, etc. The details requested are analogous to those in the previous sections.

The Report record size, number of characters that must be computed (i.e., edited for each line), the percentage of these that are alpha, its medium (in this case restricted to magnetic tape), and whether the print line format can be varied to suit the editing properties of the computer concerned, are requested. This information is designed to indicate the degree of internal processing that must be performed. It is assumed that there will be a printed line produced for each Master File Record that is updated. If any information on the size of a pre-printed form (where used) is available, it should be noted in the comments applicable to this section.

FUNCTIONAL SPECIFICATION

FUNCTIONAL SPECIFICATION

SPECIFICATION NO.	QUERY	ANSWER
FS 101	Is the application suited to magnetic tape oriented processing?	yes/no
FS 102	What is the maximum operational time within which to complete the average work load of the process described in the specification?	hours
FS 103	What is the maximum operational time within which to complete the peak work load of the process described in the specification?	hours
FS 104	What is the desirable portion of total computer operating time within which to complete the average work load of the process described in the specification?	%
FS 105	What proportion of the total essential work load is represented by the specification contained here?	% hours
FS 106	What proportion of the total possible work load is represented by the specification contained here?	% hours
FS 107	Have any other parts of the possible work load been specified in this manner for joint use for ranking purposes? If so, quote a reference.	
FS 108	What is the estimated budget allocation for the installation?	$ /month

TRANSACTION FILE

(Use 1 per Transaction File)

GENERAL

FS 201	No. of transaction records per cycle (standard).	
FS 202	No. of transaction records per cycle (peak).	
FS 203	Will the transactions be sorted in main file order?	yes/no
FS 204	Will the transactions already be on magnetic tape?	yes/no
FS 205	May the analyst alter the format of the transaction records to suit the particular system?	yes/no
FS 206	No. of transaction types in the file.	

(Describe each type individually on a separate Transaction File Record Type Sheet.)

TRANSACTION FILE RECORD TYPE

RECORD DESCRIPTION AND PROCESSING DETAILS

(Use 1 per Record Type)

GENERAL DETAILS

FS 310	No. of records of this type in the Transaction File.
FS 320	No. of characters (including alphabetic, numeric, and special characters) per record.
FS 330	No. of numeric digits per record. *
FS 340	Average number of active fields per record (an active field is one which is used or referred to during processing). **

* Assumed to be equal to (FS 320) ÷ 2, if not given.
** Assumed to be equal to FS 310, if not given.

DETAILS OF EACH SIGNIFICANT WORK PATH

In this section, each process which results from this transaction type is enumerated, and details are given to show how frequently the process is executed. The volume of processing which takes place during each execution of the process is described in terms of "Simple Update or Equivalent Operations," "Complex Update Steps or Equivalent Operations," and "Table References."

	PROCESS NAME:								
FS 350	% of records using this work path.								
FS 360	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)								
FS 370	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)								
FS 380	Average no. of decimal digits in the numeric operands used during the process.								
FS 390	No. of associated values (A) extracted from tables in execution of the work path processing, arranged by table size involved (T). (Ignore T if tables have less than 50 entries.)	A	T	A	T	A	T	A	T

MASTER FILE

(Use 1 per Master File)

GENERAL

FS 401	No. of records in the master file.	
FS 402	No. of major record types in the.file.	

(Describe each type individually on a separate Master File Record Type sheet.)

MASTER FILE RECORD TYPE

RECORD DESCRIPTION AND PROCESSING DETAILS

(Use 1 per Record Type)

GENERAL DETAILS

FS 510	No. of records of this type in the Master File.	
FS 520	No. of characters (including alphabetic, numeric, and special characters) per record.	
FS 530	No. of numeric digits per record. *	
FS 540	Average number of active fields per record (an active field is one which is used or referred to during processing). **	

* Assumed to be equal to (FS 520) ÷ 2, if not given.

** Assumed to be equal to FS 510, if not given.

DETAILS OF EACH SIGNIFICANT WORK PATH

In this section, each process which results from this record type is enumerated, and details are given to show how frequently the process is executed. The volume of processing which takes place during each execution of the process is described in terms of "Simple Update or Equivalent Operations," "Complex Update Steps or Equivalent Operations," and "Table References."

	PROCESS NAME:								
FS 550	% of Records using this work path.								
FS 560	No. of Simple Field Updates or Equivalent Operations per record. (This is the equivalent of the sum of the add/subtract and comparison operations needed to process a record.)								
FS 570	No. of Complex Field Update Steps or Equivalent Operations per record. (This is the equivalent of the sum of multiply and divide operations per record.)								
FS 580	Average no. of decimal digits in the numeric operands used during the process.								
FS 590	No. of associated values (A) extracted from tables in execution of the work path processing, arranged by table size involved (T). (Ignore T if tables have less than 50 entries.)	A	T	A	T	A	T	A	T

REPORT FILE

(Use 1 per Report File)

GENERAL

FS 601	No. of report records per cycle (standard).	
FS 602	No. of report records per cycle (peak).	
FS 603	Should the reports be sorted in main file order?	yes/no
FS 604	May the reports be placed on magnetic tape for off-line printing?	yes/no
FS 605	May the analyst alter the format of the report records to suit the particular system?	yes/no
FS 606	No. of report types in the file.	

(Describe each type individually on a separate Report File Record Type Sheet.)

REPORT FILE RECORD TYPE

RECORD DESCRIPTION AND PROCESSING DETAILS

Use 1 per Record Type

GENERAL DETAILS

FS 710	No. of records of this type in the Report File.	
FS 720	No. of characters (including alphabetic, numeric, and special characters) per record.	
FS 730	No. of printed lines per record.	
FS 740	Average number of active alphabetic fields per record. (An active field is one which is prepared or edited during processing rather than simply copied from some other document.)*	
FS 750	Average number of active numeric fields per record. (An active field is one which is prepared or edited during processing rather than simply copied from some other document.)**	

* Assumed to be equal to (FS 720) ÷ 20, if not given.
** Assumed to be equal to (FS 720) ÷ 10, if not given.

FUNCTIONAL CONVERSION ALGORITHM

FUNCTIONAL CONVERSION ALGORITHM

Transaction File — Use 1 Form Per Record Type

FUNCTIONAL VECTOR ELEMENT	INSTRUCTIONS	RESULT	COMMENT NO.
FV 1310_ * No. of records of this type	FS 310		
FV 1320_ * No. of characters per record	FS 320		
FV 1330_ * No. of numeric digits per record	FS 330 if present; (FS 320) ÷ 2 if FS 330 is not present.		
FV 1340_ * Average no. of active fields per record	FS 340 if present; FS 310 if FS 340 is not present.		
FV 1350_ * No. of Simple Updates per record	For each work path, calculate: ((FS 350) x (FS 360) x P ÷ 100); where P = ((FS 380) ÷ 10), rounded upward. The result is the sum of these values for all work paths.		
FV 1360_ * No. of Complex Update Steps per record	For each work path, calculate: ((FS 350) x (FS 370) x P ÷ 100); where P = ((FS 380) ÷ 10), rounded upward. The result is the sum of these values for all work paths.		
FV 1370_ * No. of Table Reference Steps per record	Enter the sum of the values (A x C_T) for all tables on all work paths in FS 390, where: C_T = 6 when T is 64 or less; C_T = 7 when T is 65 through 128; C_T = 8 when T is 129 through 256; C_T = 9 when T is 257 through 512; C_T = 10 when T is 513 through 1024; etc.		
FV 1380_ * Format control	Enter "1" if FS 205 is "yes." Enter "0" if FS 205 is "no."		

*Use an alphabetic suffix (A, B, C, ...) to distinguish each transaction file record type.

Master File — Use 1 Form per Record Type

FUNCTIONAL VECTOR ELEMENT	INSTRUCTIONS	RESULT	COMMENT NO.
FV 1510_ * No. of records of this type	FS 510		
FV 1520_ * No. of characters per record	FS 520		
FV 1530_ * No. of numeric digits per record	FS 530 if present; (FS 520) $\div$ 2 if FS 530 is not present.		
FV 1540_ * Average no. of active fields per record	FS 540 if present; FS 510 if FS 540 is not present.		
FV 1550_ * No. of Simple Updates per record	For each work path, calculate: ((FS 550) x (FS 560) x P $\div$ 100); where P = ((FS 580) $\div$ 10), rounded upward. The result is the sum of these values for all work paths.		
FV 1560_ * No. of Complex Update Steps per record	For each work path, calculate: ((FS 550) x (FS 570) x P $\div$ 100); where P = ((FS 580) $\div$ 10), rounded upward. The result is the sum of these values for all work paths.		
FV 1570_ * No. of Table Reference Steps per record	Enter the sum of the values ($A \times C_T$) for all tables on all work paths in FS 590, where: C_T = 6 when T is 64 or less; C_T = 7 when T is 65 through 128; C_T = 8 when T is 129 through 256; C_T = 9 when T is 257 through 512; C_T = 10 when T is 513 through 1024; etc.		

* Use an alphabetic suffix (A, B, C, ...) to distinguish each master file record type.

Report File — Use 1 Form Per Record Type

FUNCTIONAL VECTOR ELEMENT	INSTRUCTIONS	RESULT	COMMENT NO.
FV 1710_ * No. of records of this type	FS 710		
FV 1720_ * No. of characters per record	FS 720		
FV 1730_ * No. of printed lines per record	FS 730		
FV 1740_ * No. of active alphabetic fields per record	FS 740 if present; (FS 720) ÷ 20 if FS 740 is not present.		
FV 1750_ * No. of active numeric fields per record	FS 750 if present; (FS 720) ÷ 10 if FS 750 is not present		
FV 1760_ * Format control	Enter "1" if FS 605 is "yes." Enter "0" if FS 605 is "no."		

*Use a alphabetic suffic (A, B, C, . . .) to distinguish each report file record type.

FUNCTIONAL VECTOR

FUNCTIONAL VECTOR

FUNCTIONAL VECTOR ELEMENT	COMPONENT	VALUE
1310	No. of records of this type	
1320	No. of characters per record	
1330	No. of numeric digits per record	
1340	Average no. of active fields per record	
1350	No. of Simple Updates per record	
1360	No. of Complex Update Steps per record	
1370	No. of Table Reference Steps per record	
1380	Format Control	
1510	No. of records of this type	
1520	No. of characters per record	
1530	No. of numeric digits per record	
1540	Average no. of active fields per record	
1550	No. of Simple Updates per record	
1560	No. of Complex Update Steps per record	
1570	No. of Table Reference Steps per record	
1710	No. of records of this type	
1720	No. of characters per record	
1730	No. of printed lines per record	
1740	No. of active alphabetic fields per record	
1750	No. of active numeric fields per record	
1760	Format Control	

FUNCTIONAL TRANSFORMATION ALGORITHM

Pre-computation of number of FIXED NUMERIC INPUT FIELDS.

Pre-requisites: FV 1380 = 0. (If FV 1380 = 1, the value of TFV 01 is 0.)

Method

This element can be arrived at by computing the following value for each transaction file record type whose format may not be altered by the analyst:

$$(\text{No. of records of this type}) \times (\text{No. of active fields per record}) \times \left[\frac{\text{No. of numeric digits per record}}{\text{No. of characters per record}}\right]$$

The values for each record type are then added together, and their sum is the value of TFV 01.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1340, FV 1330, and FV 1320, the value of TFV 01 is the sum of the values for each transaction file record type of:

$$(\text{FV } 1310X) \times (\text{FV } 1340X) \times (\text{FV } 1330X) \div (\text{FV } 1320X),$$

where X = A, B, C, ... (the alphabetic suffix used to distinguish transaction file record types).

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

The total of these values, which is TFV 01, = __________

(Note that if FV 1380 is "1", then TFV 01 = 0.)

Pre-computation of number of FIXED ALPHAMERIC INPUT FIELDS.

Pre-requisites: FV 1380 = 0. (If FV 1380 = 1, the value of TFV 02 is 0.)

Method

This element can be arrived at by computing the following value for each transaction file record type whose format may not be altered by the analyst:

$$(\text{No. of records of this type}) \times (\text{No. of active fields per record}) \times \left[\frac{(\text{No. of characters per record}) - (\text{No. of numeric digits per record})}{\text{No. of characters per record}}\right]$$

The values for each record type are then added together, and their sum is the value of TFV 02.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1340, FV 1320, and FV 1330, the value of TFV 02 is the sum of the values for each transaction file record type of:

$$(\text{FV }1310X) \times (\text{FV }1340X) \times ((\text{FV }1320X) - (\text{FV }1330X)) \div (\text{FV }1320X),$$

where X = A, B, C, ... (the alphabetic suffix used to distinguish transaction file record types).

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

The total of these values, which is TFV 02, = __________

(Note that if FV 1380 is "1", then TFV 02 = 0.)

Pre-computation of number of FLEXIBLE NUMERIC INPUT FIELDS.

Pre-requisites: FV 1380 = 1. (If FV 1380 = 0, the value of TFV 03 is 0.)

Method

This element can be arrived at by computing the following value for each transaction file record type whose format may be altered by the analyst:

$$(\text{No. of records of this type}) \times (\text{No. of active fields per record}) \times \left[\frac{\text{No. of numeric digits per record}}{\text{No. of characters per record}}\right].$$

The values for each record type are then added together, and their sum is the value of TFV 03.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1340, FV 1330, and FV 1320, the value of TFV 03 is the sum of the values for each transaction file record type of:

$$(\text{FV 1310X}) \times (\text{FV 1340X}) \times (\text{FV 1330X}) \div (\text{FV 1320X}),$$

where X = A, B, C, . . . (the alphabetic suffix used to distinguish transaction file record types).

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

(______) x (______) x (______) ÷ (______) = __________

The total of these values, which is TFV 03, = __________.

(Note that if FV 1380 is "0", then TFV 03 = 0.)

Pre-computation of number of FLEXIBLE ALPHAMERIC INPUT FIELDS.

Pre-requisites: FV 1380 = 1. (If FV 1380 = 0, the value of TFV 04 is 0.)

Method

This element can be arrived at by computing the following value for each transaction file record type whose format may be altered by the analyst:

$$(\text{No. of records of this type}) \times (\text{No. of active fields per record}) \times \left[\frac{(\text{No. of characters per record}) - (\text{No. of numeric digits per record})}{\text{No. of characters per record}}\right]$$

The values for each record type are then added together, and their sum is the value of TFV 04.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1340, FV 1320, and FV 1330, the value of TFV 04 is the sum of the values for each transaction file record type of:

$$(\text{FV 1310X}) \times (\text{FV 1340X}) \times ((\text{FV 1320X}) - (\text{FV 1330X})) \div (\text{FV 1320X}),$$

where X = A, B, C, ... (the alphabetic suffix used to distinguish transaction file record types).

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

(______) x (______) x ((______) - (______)) ÷ (______) = __________

The total of these values, which is TFV 04, = __________

(Note that if FV 1380 is "0", then TFV 04 = 0.)

Pre-computation of number of SIMPLE UPDATE STEPS.

<u>Pre-requisites:</u> None.

<u>Method</u>

This element can be arrived at by computing the following value for each transaction file and each master file record type:

(No. of records of this type) x (No. of Simple Updates per record).

The values for each record type are then added together, and their sum is the value of TFV 05.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1350, FV 1510, and FV 1550, the value of TFV 05 is the <u>sum</u> of the values for each transaction file record type of:

(FV 1310X) x (FV 1350X)

and the values for each master file record type of:

(FV 1510X) x (FV 1550X);

where X = A, B, C ..., depending upon the number of record types in each file.

(________) x (________) = ____________

(________) x (________) = ____________

(________) x (________) = ____________

(________) x (________) = ____________

(________) x (________) = ____________

(________) x (________) = ____________

The total, which is TFV 05, = ____________

Pre-computation of number of COMPLEX UPDATE STEPS.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each transaction file and each master file record type:

(No. of records of this type) x (No. of Complex Update Steps per record).

The values for each record type are then added together, and their sum is the value of TFV 06.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1360, FV 1510, and FV 1560, the value of TFV 06 is the sum of the values for each transaction file record type of:

(FV 1310X) x (FV 1360X)

and the values for each master file record type of:

(FV 1510X) x (FV 1560X);

where X = A, B, C ..., depending upon the number of record types in each file.

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 06, = __________.

Pre-computation of number of TABLE REFERENCE STEPS.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each transaction file and each master file record type:

(No. of records of this type) x (No. of Table Reference Steps per record).

The values for each record type are then added together, and their sum is the value of TFV 07.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1350, FV 1510, and FV 1550, the value of TFV 07 is the sum of the values for each transaction file record type of:

(FV 1310X) x (FV 1370X)

and the values for each master file record type of:

(FV 1510X) x (FV 1570X);

where X = A, B, C . . . , depending upon the number of record types in each file.

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 07, = __________.

Pre-computation of number of FIXED NUMERIC OUTPUT FIELDS.

Pre-requisites: FV 1760 = 0. (If FV 1760 = 1, the value of TFV 08 is 0.)

Method

This element can be arrived at by computing the following value for each report file record type whose format may not be altered by the analyst:

(No. of records of this type) x (No. of active numeric fields per record).

The values for each record type are then added together, and their sum is the value of TFV 08.

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1750, the value of TFV 08 is the sum of the values for each report file record type of:

(FV 1710X) x (FV 1750X),

where X = A, B, C, ...

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 08, = __________

(Note that if FV 1760 is "1", then TFV 08 = 0.)

Pre-computation of number of FIXED ALPHAMERIC OUTPUT FIELDS.

Pre-requisites: FV 1760 = 0. (If FV 1760 = 1, the value of TFV 09 is 0.)

Method

This element can be arrived at by computing the following value for each report file record type whose format may not be altered by the analyst:

(No. of records of this type) x (No. of active alphameric fields per record).

The values for each record type are then added together, and their sum is the value of TFV 09.

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1740, the value of TFV 09 is the sum of the values for each report file record type of:

(FV 1710X) x (FV 1740X),

where X = A, B, C, ...

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 09, = __________.

(Note that if FV 1760 is "1", then TFV 09 = 0.)

Pre-computation of number of FLEXIBLE NUMERIC OUTPUT FIELDS.

Pre-requisites: FV 1760 = 1. (If FV 1760 = 0, the value of TFV 10 is 0.)

Method

This element can be arrived at by computing the following value for each report file record type whose format may be altered by the analyst:

(No. of records of this type) x (No. of active numeric fields per record).

The values for each record type are then added together, and their sum is the value of TFV 10.

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1750, the value of TFV 10 is the sum of the values for each report file record type of:

(FV 1710X) x (FV 1750X),

where X = A, B, C, ...

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 10, = __________.

(Note that if FV 1760 is "0", then TFV 10 = 0.)

Pre-computation of number of FLEXIBLE ALPHAMERIC OUTPUT FIELDS.

<u>Pre-requisites</u>: FV 1760 = 1. (If FV 1760 = 0, the value of TFV 11 is 0.)

<u>Method</u>

This element can be arrived at by computing the following value for each report file record type whose format may be altered by the analyst:

(No. of records of this type) x (No. of active alphameric fields per record).

The values for each record type are then added together, and their sum is the value of TFV 11.

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1740, the value of TFV 11 is the <u>sum</u> of the values for each report file record type of:

(FV 1710X) x (FV 1740X),

where X = A, B, C, ...

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 11, = __________.

(Note that if FV 1760 is "0", then TFV 11 = 0.)

Pre-computation of number of RECORDS IN ALL FILES.

Pre-requisites: None.

Method

This element can be arrived at by summing up the number of records of each transaction file, master file, and report file record type.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1510, and FV 1710, the value of TFV 12 is the sum of the number of records of each type:

(FV 1310A) + (FV 1310B) + (FV 1310C) + ...

\+ (FV 1510A) + (FV 1510B) + (FV 1510C) + ...

\+ (FV 1710A) + (FV 1710B) + (FV 1710C) + ...;

i.e., (________) + (________) + (________) + (________)

\+ (________) + (________) + (________) + (________)

\+ (________) + (________) + (________) + (________).

The total, which is TFV 12, = __________.

Pre-computation of number of CHARACTERS IN ALL FILES.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each transaction file, master file, and report file record type:

(No. of records of this type) x (No. of characters per record).

The value for each record type are then added together, and their sum is the value of TFV 13.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1320, FV 1510, FV 1520, FV 1710, and FV 1720, the value of TFV 13 is the sum of the values for each transaction file record type of:

(FV 1310X) x (FV 1320X),

and the value for each master file record type of:

(FV 1510X) x (FV 1520X),

and the value for each report file record type of:

(FV 1710X) x (FV 1720X);

where X = A, B, C, ..., depending upon the number of record types in each file.

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 13, = __________

Pre-computation of number of ALPHAMERIC CHARACTERS TRANSFERRED IN OR OUT.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each transaction file, master file, and report file record type:

(No. of records of this type) x
((No. of characters per record) - (No. of numeric digits per record)).

The transaction file, which consists of card images, and the report file, which consists of line images, are treated as completely alphameric. The master file is both read and written, so the corresponding values are doubled.

The values for each record type are then added together, and their sum is the value of TFV 14.

As these values are listed separately for each record type within the Functional Vector as FV 1310, FV 1320, FV 1510, FV 1520, FV 1530, FV 1710, and FV 1720, the value of TFV 14 is the sum of the values for each transaction file record type of:

(FV 1310X) x (FV 1320X),

and the values for each master file record type of:

2 x (FV 1510X) x ((FV 1520X) - (FV 1530X)),

and the values for each report file record type of:

(FV 1710X) x (FV 1720X);

where X = A, B, C, . . . , depending upon the number of record types in each file.

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

2 x (________) x ((________) - (________)) = __________

2 x (________) x ((________) - (________)) = __________

2 x (________) x ((________) - (________)) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 14, = __________

Pre-computation of number of NUMERIC DIGITS TRANSFERRED IN OR OUT.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each master file record type:

2 x (No. of records of this type) x (No. of numeric digits per record).

The factor of "2" is applied because the master file is both read and written.

The values for each record type are then added together, and their sum is the value of TFV 15.

As these values are listed separately for each record type within the Functional Vector as FV 1510, and FV 1530, the value of TFV 15 is the sum of the values for each master file record type of:

2 x (FV 1510X) x (FV 1530X)

where X = A, B, C, . . .

2 x (______) x (______) = ________

2 x (______) x (______) = ________

2 x (______) x (______) = ________

2 x (______) x (______) = ________

The total, which is TFV 15, = ________

Pre-computation of number of CARD IMAGES TRANSFERRED IN.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each transaction file record type:

(No. of records of this type) x [(No. of characters per record) ÷ 80]*.

The values for each record type are then added together, and their sum is the value of TFV 16.

As these values are listed separately for each record type within the Functional Vector as FV 1310 and FV 1320, the value of TFV 16 is the sum of the values for each transaction file record type of:

(FV 1310X) x [(FV 1320X) ÷ 80]*,

where X = A, B, C, ...

(________) x [(________) ÷ 80]* = __________

(________) x [(________) ÷ 80]* = __________

(________) x [(________) ÷ 80]* = __________

(________) x [(________) ÷ 80]* = __________

The total, which is TFV 16, = __________

* The quantity within brackets, if not an integer, should be rounded upward to the next higher integer.

Pre-computation of number of LINE IMAGES TRANSFERRED OUT.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each report file record type:

(No. of records of this type) x [(No. of characters per record) ÷ 120]*.

The values for each record type are then added together, and their sum is the value of TFV 17.

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1720, the value of TFV 17 is the sum of the values for each report file record type of:

(FV 1710X) x [(FV 1720X) ÷ 120]* ,

where X = A, B, C, ...

(________) x [(________) ÷ 120]* = __________

(________) x [(________) ÷ 120]* = __________

(________) x [(________) ÷ 120]* = __________

(________) x [(________) ÷ 120]* = __________

The total, which is TFV 17, = __________

* The quantity within brackets, if not an integer, should be rounded upward to the next higher integer.

Pre-computation of number of ALPHAMERIC CHARACTERS IN THE MASTER FILE.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each master file record type:

(No. of records of this type) x
((No. of characters per record) - (No. of numeric digits per record)).

The values for each record type are then added together, and their sum is the value of TFV 18 (M).

As these values are listed separately for each record type within the Functional Vector as FV 1510, FV 1520, and FV 1530, the value of TFV 18 (M) is the sum of the values for each master file record type of:

(FV 1510X) x ((FV 1520X) - (FV 1530X)),

where X = A, B, C, ...

(________) x ((________) - (________)) = __________

(________) x ((________) - (________)) = __________

(________) x ((________) - (________)) = __________

(________) x ((________) - (________)) = __________

The total, which is TFV 18 (M), = __________

Pre-computation of number of ALPHAMERIC CHARACTERS IN THE REPORT FILE.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each report file record type:

(No. of records of this type) x (No. of printed lines per record) x 120.

The values for each record type are then added together, and their sum is the value of TFV 18(R).

As these values are listed separately for each record type within the Functional Vector as FV 1710 and FV 1730, the value of TFV 18(R) is the sum of the values for each report file record type of:

(FV 1710X) x (FV 1730X) x 120,

where X = A, B, C, ...

(________) x (________) x 120 = __________

(________) x (________) x 120 = __________

(________) x (________) x 120 = __________

(________) x (________) x 120 = __________

The total, which is TFV 18(R), = __________

Pre-computation of number of NUMERIC DIGITS IN THE MASTER FILE.

Pre-requisites: None.

Method

This element can be arrived at by computing the following value for each master file record type:

(No. of records of this type) x (No. of numeric digits per record).

The values for each record type are then added together, and their sum is the value of TFV 19.

As these values are listed separately for each record type within the Functional Vector as FV 1510 and FV 1530, the value of TFV 19 is the sum of the values for each master file record type of:

(FV 1510X) x (FV 1530X),

where X = A, B, C, ...

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

(________) x (________) = __________

The total, which is TFV 19, = __________

Pre-computation of number of CARD IMAGES IN THE TRANSACTION FILE.

Pre-requisites: None.

Method

This element can be arrived at by noting that, in the present version of the model, TFV 20 is identical with TFV 16 because only one transaction file can be accommodated.

Therefore, TFV 20 = TFV 16 = __________.

Pre-computation of number of LINE IMAGES IN THE REPORT FILE.

Pre-requisites: None.

Method

This element can be arrived at by noting that, in the present version of the model, TFV 21 is identical with TFV 17 because only one report file can be accommodated.

Therefore, TFV 21 = TFV 17 = __________.

TRANSFORMED FUNCTIONAL VECTOR

TRANSFORMED FUNCTIONAL VECTOR

TRANSFORMED FUNCTIONAL VECTOR ELEMENT	COMPONENT	VALUE
01	Number of fixed numeric input fields.	
02	Number of fixed alphameric input fields.	
03	Number of flexible numeric input fields.	
04	Number of flexible alphameric input fields.	
05	Number of "Simple Update" steps.	
06	Number of "Complex Update" steps.	
07	Number of "Table Reference" steps.	
08	Number of fixed numeric output fields.	
09	Number of fixed alphameric output fields.	
10	Number of flexible numeric output fields.	
11	Number of flexible alphameric output fields.	
12	Number of records in all files.	
13	Number of characters in all files.	
14	Number of alphameric characters transferred in or out.	
15	Number of numeric characters transferred in or out.	
16	Number of card images transferred in and out.	
17	Number of line images transferred out.	

TRANSFORMED FUNCTIONAL VECTOR ELEMENT	COMPONENT	VALUE BY FILE			
18	Number of alphameric characters (by file).				
19	Number of numeric characters (by file).				
20	Number of card images (by file).				
21	Number of line images (by file)				

APPENDIX III

GUIDE AND FORMS FOR COMPLETION OF PERFORMANCE ALGORITHMS OF THE "VECTOR" ESTIMATING PROCESS

1. INTRODUCTION

The VECTOR method uses independent sets of numbers (Vectors) to represent the important characteristics of each computer and each application. The Vectors are used both to establish and time the optimum method on a specific computer system. The Vectors themselves are normally self-sufficient and no other knowledge of either the problem or the computer is assumed.

In these circumstances it can be seen that the creation of these Vectors is an unusually responsible process. This document is concerned with the preparation of the Performance Estimate, shows the procedural steps needed, and attempts to illustrate the requirements and the opportunities given to allow each computer system to be shown at its honest best.

GUIDE TO COMPLETION

2. GUIDE FOR EXECUTING THE COMPUTATION PROCESS

For this purpose, a set of blanks have been supplied. These may be entirely blank (as Figure 1) and are filled directly from the Transformed Engineering and Transformed Functional Vectors (Figure 2 and 3) which provide all the detail needed for the estimation process. Each quantity from the Transformed Functional Vector (entered in the left hand columns) is multiplied by each of four values of the Transformed Engineering Vector, giving a value in microseconds which is then rounded to the nearest second before entering into the appropriate position in the right hand columns depending on the specific constraint of the Engineering Vector. (See Figure 4.)

PERFORMANCE ALGORITHM

Central Processor Times

TFV Description	TFV Value	TEV Description	TEV Value, µsec		RESULTS: (TFV Value) x (TEV Value)*			
					General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 01 No. of fixed numeric input fields		TEV 01 Edit a fixed numeric field during input	G					
			CP					
			I-O					
			S					
TFV 02 No. of fixed alphameric input fields		TEV 02 Edit a fixed alphameric field during input	G					
			CP					
			I-O					
			S					
TFV 03 of flexible		TEV 03 Edit a flexible	G					
			CP					

Figure 1. Computation Worksheet Blanks

TRANSFORMED FUNCTIONAL VECTOR

TRANSFORMED FUNCTIONAL VECTOR ELEMENT	COMPONENT	VALUE
01	Number of fixed numeric input fields.	
02	Number of fixed alphameric input fields.	
03	Number of flexible numeric input fields.	
04	Number of flexible alphameric input fields.	
05	Number of "Simple Update" steps.	
06	Number of "Complex Update" steps.	
07	Number of "Table Reference" steps.	
08	Number of fixed numeric output fields.	
09	Number of fixed alphameric output fields.	
10	Number of flexible numeric output fields.	
11	Number of flexible alphameric output fields.	
12	Number of records in all files.	
13	Number of characters in all files.	
14	Number of alphameric characters transferred in or out.	
15	Number of numeric characters transferred in or out.	
16	Number of card images transferred in and out.	
17	Number of line images transferred out.	

TRANSFORMED FUNCTIONAL VECTOR ELEMENT	COMPONENT	VALUE BY FILE			
18	Number of alphameric characters (by file).				
19	Number of numeric characters (by file).				
20	Number of card images (by file).				
21	Number of line images (by file).				

Figure 2. Transformed Functional Vector

TRANSFORMED ENGINEERING VECTOR

TRANSFORMED ENGINEERING VECTOR ELEMENT	COMPONENT	VALUE UNDER GENERAL AND SPECIFIC LIMITING FACTORS			
		GEN	CP	I/O	SPACE
01	Edit a fixed numeric field during input				
02	Edit a fixed alphameric field during input				
03	Edit a flexible numeric field during input				
04	Edit a flexible alphameric field during input				
05	Simple Update Operation				
06	Complex Update				
07	Table Reference Time				
08	Edit a fixed numeric field during output				
09	Edit a fixed alphameric field during output				
10	Edit a flexible numeric field during output				
11	Edit a flexible alphameric field during output				
12	Control the processing of a record				
13	Control the movement of a record				
14	Load on central processor per alphameric character				
15	Load on central processor per decimal digit				
16	Load on central processor per card image				
17	Load on central processor per line image				
18	Magnetic tape performance on alphameric data				
19	Magnetic tape performance on decimal data				
20	Magnetic tape performance on card images				
21	Magnetic tape performance on line images				
22	Simultaneity rule number				
23	Simultaneity parameter number				

Figure 3 Transformed Engineering Vector

PERFORMANCE ALGORITHM

PERFORMANCE ALGORITHM

Central Processor Times

TFV Description	TFV Value	TEV Description	TEV Value, µsec		RESULTS: (TFV Value) x (TEV Value)*			
					General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 01 No. of fixed numeric input fields		TEV 01 Edit a fixed numeric field during input	G	3,430	.34			
			CP	460		4.6		
			I-O	3,430			34	
			S	3,430				34
TFV 02 No. of fixed alphameric input fields		TEV 02 Edit a fixed alphameric field during input	G	3,687	18.4			
			CP	295		1.48		
			I-O	3,687			184	
			S	3,687				184
TFV 03 ...of flexible		TEV 03 Edit a flexible	G					
			CP					

Figure 4. Computation Worksheet Blanks

This then provides four times for each item, one under no known constraint, and three for I/O, C.P., and Space being respectively the limiting quantities. These are totaled in two places, one on the third page for the central processor (Figure 5) and then one on each "file" page. These totals are brought forward entered on the System Timing Sheet (Figure 6) and grouped under the Simultaneity Rules and Parameter shown in the Transformed Engineering Vector, and provided with the Computation Worksheets.

		...processor per card image	I-O					
			S					
TFV 17 No. of line images transferred out		TEV 17 Load on central processor per line image	G					
			CP					
			I-O					
			S					
Subtotals from first page:								
Subtotals from second page:								
TOTAL CENTRAL PROCESSOR TIMES:								

* Round each result to the nearest second.

Figure 5. Central Processor Totals To Be Carried Forward

PERFORMANCE ALGORITHM

Central Processor Times

TFV Description	TFV Value	TEV Description	TEV Value, μsec		RESULTS: (TFV Value) x (TEV Value)* General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 01 No. of fixed numeric input fields		TEV 01 Edit a fixed numeric field during input	G					
			CP					
			I-O					
			S					
TFV 02 No. of fixed alphameric input fields		TEV 02 Edit a fixed alphameric field during input	G					
			CP					
			I-O					
			S					
TFV 03 No. of flexible numeric input fields		TEV 03 Edit a flexible numeric field during input	G					
			CP					
			I-O					
			S					
TFV 04 No. of flexible alphameric input fields		TEV 04 Edit a flexible alphameric field during input	G					
			CP					
			I-O					
			S					
TFV 05 No. of Simple Update Operations		TEV 05 Simple Update Operation	G					
			CP					
			I-O					
			S					
TFV 06 No. of Complex Update Steps		TEV 06 Complex Update Step	G					
			CP					
			I-O					
			S					
Subtotals (to be carried to third page):								

*Round each result to the nearest second.

TFV Description	TFV Value	TEV Description	TEV Value, μsec		RESULTS: (TFV Value) x (TEV Value)*			
					General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 07 No. of Table Reference Steps		TEV 07 Table Reference Time	G					
			CP					
			I-O					
			S					
TFV 08 No. of fixed numeric output fields		TEV 08 Edit a fixed numeric field during output	G					
			CP					
			I-O					
			S					
TFV 09 No. of fixed alphameric output fields		TEV 09 Edit a fixed alphameric field during output	G					
			CP					
			I-O					
			S					
TFV 10 No. of flexible numeric output fields		TEV 10 Edit a flexible numeric field during output	G					
			CP					
			I-O					
			S					
TFV 11 No. of flexible alphameric output fields		TEV 11 Edit a flexible alphameric field during output	G					
			CP					
			I-O					
			S					
TEV 12 No. of records in all files		TEV 12 Control the processing of a record	G					
			CP					
			I-O					
			S					
Subtotals (to be carried to third page):								

* Round each result to the nearest second.

TFV Description	TFV Value	TEV Description	TEV Value, μsec		RESULTS: (TFV Value) x (TEV Value)*			
					General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 13 No. of characters in all files		TEV 13 Control the movement of a record (per character)	G					
			CP					
			I-O					
			S					
TFV 14 No. of alphameric characters transferred in or out		TFV 14 Load on central processor per alphameric character	G					
			CP					
			I-O					
			S					
TFV 15 No. of decimal digits transferred in or out		TEV 15 Load on central processor per decimal digit	G					
			CP					
			I-O					
			S					
TFV 16 No. of card images transferred in		TEV 16 Load on central processor per card image	G					
			CP					
			I-O					
			S					
TFV 17 No. of line images transferred out		TEV 17 Load on central processor per line image	G					
			CP					
			I-O					
			S					
Subtotals from first page:								
Subtotals from second page:								
TOTAL CENTRAL PROCESSOR TIMES:								

* Round each result to the nearest second.

File Times (Use one form for each file)

FILE NAME: ____________________

TFV Description	TFV Value	TEV Description	TEV Value, μsec		RESULTS: (TFV Value) x (TEV Value)* General Conditions	If CP Critical	If I-O Critical	If Space Critical
TFV 18 No. of alphameric characters (Enter "0" for transaction and report files)		TEV 18 Magnetic tape performance on alphameric data	G					
			CP					
			I-O					
			S					
TFV 19 No. of decimal digits (Enter "0" for transaction and report files)		TEV 19 Magnetic tape performance on decimal data	G					
			CP					
			I-O					
			S					
TFV 20 No. of card images (Enter "0" for master and report files)		TEV 20 Magnetic tape performance on card images	G					
			CP					
			I-O					
			S					
TFV 21 No. of line images (Enter "0" for transaction and master files)		TEV 21 Magnetic tape performance on line images	G					
			CP					
			I-O					
			S					
TOTAL TIMES FOR ____________ FILE:								

SIMULTANEITY RULE

SIMULTANEITY RULE NUMBER 1

If a computer has a Simultaneity rule number 1, it indicates that the reading and writing channels of the system are separate, and that internal processing inhibits either an input or an output channel.

The parameter specifies the number of input channels; there are an equivalent number of output channels.

From the preceding sections, the time involved in processing each of the input and output files and the central processor time are known. These times (including the central processor time) should be arranged between the input channels and output channels so as to minimize the time used by the most time-consuming channel. For this purpose, the central processor time may be split between two or more channels, but the time for an individual file may not be split.

The time utilized by the most time-consuming channel, when this has been minimized, is the total time for the application under one condition. By taking in turn the times in the previous sections for each of the four conditions, and minimizing them, the lowest of these times (which is the performance time) can be found.

SIMULTANEITY RULE NUMBER 2

If a computer has a Simultaneity rule number 2, it indicates that the reading and writing channels of the system are separate, and that internal processing does not inhibit an input or an output channel.

The parameter specifies the number of input channels; there are an equivalent number of output channels.

From the preceding sections, the time involved in processing each of the input and output files and the central processor time are known. The input and output times should be arranged between the input channels and output channels so as to minimize the time used by the most time-consuming channel. Central processor time should be considered as a separate "channel." File times must be treated as blocks and cannot be split between channels.

The time utilized by the most time-consuming channel, when this has been minimized, is the total time for the application under one condition. By taking in turn the times in the previous sections for each of the four conditions, and minimizing them, the lowest of these times (which is the performance time) can be found.

SIMULTANEITY RULE NUMBER 3

If a computer has a Simultaneity rule number 3, it indicates that the reading and writing channels of the system are not separate, and that internal processing inhibits a channel.

The parameter specifies the number of channels.

From the preceding section, the time involved in processing each of the input and output files and the central processor time are known. These times (including the central processor time) should be arranged between the channels so as to minimize the time used by the most time-consuming channel. For this purpose, the central processor time may be split between two or more channels, but the time for an individual file may not be split.

The time utilized by the most time-consuming channel, when this has been minimized, is the total time for the application under one condition. By taking in turn the times in the previous section for each of the four conditions, and minimizing them, the lowest of these times (which is the performance time) can be found.

SIMULTANEITY RULE NUMBER 4

If a computer has a Simultaneity rule number 4, it indicates that the reading and writing channels of the system are not separate, and that internal processing does not inhibit a channel.

The parameter specifies the number of channels.

From the preceding section, the time involved in processing each of the input and output files and the central processor time are known. The input and output times should be arranged between the channels so as to minimize the time used by the most time-consuming channel. Central processor time should be considered as a separate "channel." File times must be treated as blocks and cannot be split between channels.

The time utilized by the most time-consuming channel, when this has been minimized, is the total time for the application under one condition. By taking in turn the times in the previous sections for each of the four conditions, and minimizing them, the lowest of these times (which is the performance time) can be found.

SYSTEMS TIMING WORK SHEET

SYSTEM TIMING WORKSHEET

(To be prepared before use in accordance with specific Simultaneity Rules and Parameters)

Times, in Seconds, Under General Conditions

		Ch. 1 CP/I/O	Ch. 2 CP/I/O	Ch 3 CP/I/O	Ch. 4 CP/I/O	Ch. 5 CP/I/O
Central Processor	-					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
Time Utilized In Each Channel						

System Time, Under General Conditions: ________ Seconds.

Times, In Seconds, Under Central Processor Limited Conditions

		Ch. 1 CP/I/O	Ch. 2 CP/I/O	Ch 3 CP/I/O	Ch. 4 CP/I/O	Ch. 5 CP/I/O
Central Processor	-					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
Time Utilized In Each Channel						

System Time, Under Central Processor Limited Conditions ________ Seconds.

Times, In Seconds, Under Input/ Output Limited Conditions

		Ch. 1 CP/I/O	Ch. 2 CP/I/O	Ch 3 CP/I/O	Ch. 4 CP/I/O	Ch. 5 CP/I/O
Central Processor	-					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
Time Utilized In Each Channel						

System Time, Under Input/Output Limited Conditions: ________ Seconds.

Times, In Seconds, Under Space Limited Conditions

		Ch. 1 CP/I/O	Ch. 2 CP/I/O	Ch 3 CP/I/O	Ch. 4 CP/I/O	Ch. 5 CP/I/O
Central Processor	-					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
________ File	In/Out					
Time Utilized In Each Channel						

System Time, Under Space Limited Conditions: ________ Seconds.

The minimum system time is ________ seconds, or ________ hours and ________ minutes. This is the time required for the ________ computer system on the ________ problem.